CSC 148: Introduction to Computer Science

Week 2

Object-Oriented Programming

Reminder: MUST to do the readings **before lecture** !

In class: apply content in exercises, discuss, ask questions

=> develop stronger command of the concepts!

University of Toronto Mississauga,

Department of Mathematical and Computational Sciences

The fundamental transition

- The study of computer science transforms us from being users of technology to being creators of technology

- As a computer scientist, you will have to design programs according to *your client's* specifications

- Skills to learn: Design (listening!), implementation, writing (good!) documentation, (thorough!) testing

Classes and objects

- What’s a class?
  - Abstract data structure that models a real-world concept
  - Describes the attributes and “abilities” (methods) of that concept (called object)
  - Example: int, str, list, etc., or user-defined: Tweet, User, Cat, Desk, FileReader, ColourPrinter, etc.

- What’s an object?
  - Instance of a class
  - Everything in Python is an object!

Design roadmap: Analyze the specs

- Analyze specification:

  The Twitter (X) application allows users to broadcast short messages called tweets. A tweet includes the message content (of up to 280 characters), the user who wrote the tweet, when the tweet was created, and how many "likes" the tweet has. Once a tweet is created, it may be liked by other users. Furthermore, the tweet may be edited by its owner.

Design roadmap: Analyze the specs

- Analyze specification:

  The Twitter (X) application allows users to broadcast short messages called ***tweets****. A* ***tweet*** *includes the message content (of up to 280 characters), the user who wrote the* ***tweet****, when the* ***tweet*** *was created, and how many "likes" the* ***tweet*** *has. Once a* ***tweet*** *is created, it may be liked by other users. Furthermore, the* ***tweet*** *may be edited by its owner.*

Design roadmap: Analyze the specs

- Analyze specification:

  The Twitter (X) application allows users to broadcast short messages called **tweets**. A **tweet** includes the message **content** (of up to 280 characters), the **user** who wrote the **tweet**, **when** the **tweet** was **created**, and how many “**likes**” the **tweet** has. Once a **tweet** is created, it may be liked by other users. Furthermore, the **tweet** may be edited by its owner.

Design roadmap: Analyze the specs

- Analyze specification:

  The Twitter (X) application allows users to broadcast short messages called **tweets**. A **tweet** includes the message **content** (of up to 280 characters), the **user** who wrote the **tweet**, **when** the **tweet** was **created**, and how many "**likes**" the **tweet** has. Once a **tweet** is created, it may be **liked** by other users. Furthermore, the **tweet** may be **edited** by its owner.

Design roadmap: Define the API

- Define the class API according to the class design recipe:
  - 1. Class name and brief description (docstring)
  - 2. Examples of client code (as doctests in the class docstring)
  - 3. Public methods
    - Apply the Function design recipe!
  - 4. Public attributes

Design roadmap: Implement the class

- Implement the class
  - 5. Internal (private) attributes
  - 6. Representation invariants
  - 7. Implement public methods
- We'll revisit privateness and representation invariants later, but keep them in mind

Worksheet ...

- Reminder
  - Read carefully what the question asks
  - Formulate answers on your own
  - Discuss with neighbours
  - Ask questions along the way

Rebinding self

```
def mutate(self, x):
    self = NewObject(x)
```

Rebinding `self` doesn't mutate anything!

Learning Tips: What to do after lecture

- **Review**: summarize, question, re-explain
- **Share**: meet with a friend or study group
- **Get help**: come to office hours!