

What does it mean for an obj to be an instance of class?

→ the object has that class type

What does "self" represent?

→ specific instance

→ an object that we call a method on

What are member variables?

→ data in the object

→ each instance has their own copy

→ accessible for all methods

→ aka instance variables

What are methods?

→ functions in a class

What is OOP?

→ Object Oriented Programming

→ programs define classes

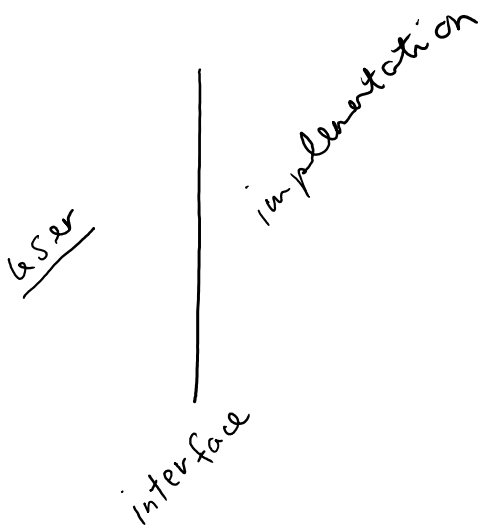
→ nouns in problem are classes;
verbs are functions

Program design
matches our
mental model
of the problem

Interfaces vs. Implementations

Interfaces vs. Implementations

→ keep strict separation



interface is contract
between user + implementation

ex. interface of a fn is its
inputs + outputs
(params) (return values)

interface of a class is
its set of methods

Advantages: user doesn't need to know how
a fn works to use it & they
don't need to know how class stores
data or implements methods to use it

implementation need to know how the
user will use the function

- user might be
 - on other class
 - a function
 - a human

EX

We define setters & getters to shield
the user from how we store data in
the Snack class

→ changing the name or type of
instance variables doesn't
require the user to change
their code