LEC 10

CSE 122

Introduction to Objects

BEFORE WE START

Talk to your neighbors:
Favorite study spot on campus?

Music: Miya's 23wi CSE 122 Playlist

Instructor Miya Natsuhara

TAs

Ayush	Atharva	Ernie	Ambika
Connor	Julia	Di	Elizabeth
Poojitha	Megana	Logan	Joe
Andrew A	Joey	Shivani	Jin
Andrew C	Eesha	Michelle	Ben
Jasmine	Lilian	Steven	Evelyn
Darel	Thomas	Kevin	Kent
Gabe	Leon	Ken	
Karen	Melissa	Vivek	
Colton	Audrey	Autumn	

Questions during Class?

Raise hand or send here

sli.do #cse122

Lecture Outline

- Announcements
- SearchEngine Recap
- Bias in Data Discussion
- OOP Review
- Example
- Abstraction

Announcements

- Programming Assignment 2 due date pushed one week!
  - Due Thursday, Feb 16
- No assignment released on Friday!
- Quiz 1 Retakes starting next week
  - Quiz 1 and Quiz 0 Retake grades coming out soon

Lecture Outline

- Announcements
- SearchEngine Recap
- Bias in Data Discussion
- OOP Review
- Example
- Abstraction

Lecture Outline

- Announcements
- SearchEngine Recap
- **Bias in Data Discussion**
- OOP Review
- Example
- Abstraction

Data Bias

- Common Misconception: Models or Artificial Intelligence (AI) are somehow “less biased” or “more objective” than humans. **Not true.**
- The programs we use operate on real-world data, and will often reflect the biases that data contains
- Have to carefully consider the context and limitations of the data we gather. If the data an algorithm is built on is vastly different than the context in which it’s used, some pretty awful outcomes can happen

Data Bias

In modern artificial intelligence, data rules. A.I. software is only as smart as the data used to train it. If there are many more white men than black women in the system, it will be worse at identifying the black women.

One widely used facial-recognition data set was estimated to be more than 75 percent male and more than 80 percent white, according to another research study.

Color Matters in Computer Vision

Facial recognition algorithms made by Microsoft, IBM and Face++ were more likely to misidentify the gender of black women than white men.

Gender was misidentified in **up to 1 percent of lighter-skinned males** in a set of 385 photos.

Gender was misidentified in **up to 7 percent of lighter-skinned females** in a set of

Gender was misidentified in **up to 12 percent of darker-skinned males** in a set of 318 photos.

Gender was misidentified in **35 percent of darker-skinned females** in a set of 271 photos.

What to do?

- Obviously, ideal to have datasets that aren't biased in the first place.
  - But might not always be possible if we can't fix the sources of the bias in the real world...
- AI/Models aren't "neutral" or "more objective", they just quickly and automatically codify the status quo (and perpetuate biases)
  - Garbage in → Garbage out
- Lots of work going into how to de-bias models *even if* they are trained on biased data. Active area of research!
  - Key take-away: None of this comes "for free", requires hard work to fight bias
- Ask ourselves:
  - What biases might be present in my data?
  - What assumptions might I be making about who is using my program?
  - How can I write code to be more inclusive?
  - What happens *when (**not if**)* mistakes happen? Who potentially benefits and who is potentially harmed?

Lecture Outline

- Announcements
- SearchEngine Recap
- Bias in Data Discussion
- **OOP Review** 
- Example
- Abstraction

(PCM) Object Oriented Programming (OOP)

- **procedural programming**: Programs that perform their behavior as a series of steps to be carried out
  - Classes that *do* things

- **object-oriented programming (OOP):** Programs that perform their behavior as interactions between objects
  - Classes that *represent* things
  - We're going to start writing our own objects!

(PCM) Classes & Objects

- Classes can define the *template* for an object
  -  Like the blueprint for a house!

- Objects are the actual *instances* of the class
  -  Like the actual house built from the blueprint!

We create a new instance of a class with the `new` keyword

e.g., `Scanner console = new Scanner(System.in);`

(PCM) State & Behavior

- Objects can tie related *state* and *behavior* together

- *State* is defined by the object's *fields* or *instance variables*
  - *Scanner's state may include what it's scanning, where it is in the input, etc.*

- *Behavior* is defined by the object's *instance methods*
  - *Scanner's behavior includes "getting the next token and returning it as an `int`", "returning whether there is a next token or not", etc.*

(PCM) Syntax

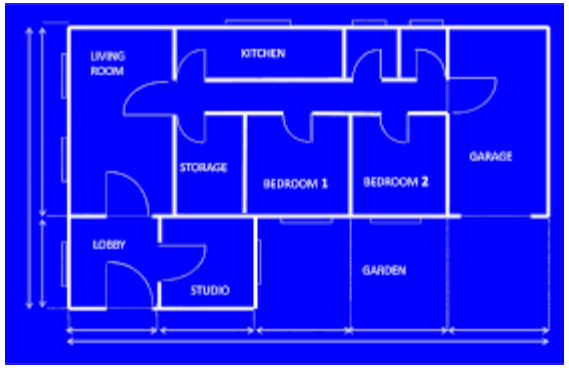
```
public class MyObject {
    // fields
    type1 fieldName1;
    type2 fieldName2;
    ...

    // instance methods
    public returnType methodName(...) {
        ...
    }
}
```

(PCM) Instance Variables

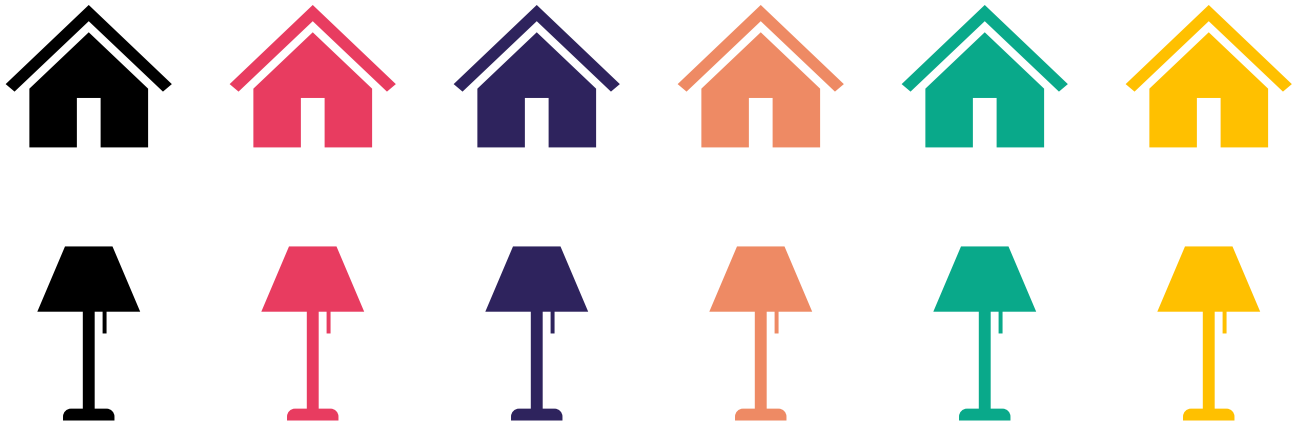
- Fields are also referred to as *instance variables*

- Fields are defined in a class
- Each *instance* of the class has their own copy of the fields
  - Hence *instance* variable! It's a variable tied to a specific instance of the class!



(PCM) Instance Methods

- Instance methods are defined in a class
- Calling an instance method on a particular *instance* of the class will have effects on *that* instance



Lecture Outline

- Announcements
- SearchEngine Recap
- Bias in Data Discussion
- OOP Review
- **Example**
- Abstraction

Representing a point

How would we do this given what we knew last week?

Maybe `int x, int y`?

Maybe `int[]`?

Representing a point

`int x, int y`

- Easy to mix up x, y
- Just two random `int`s floating around – easy to make mistakes!

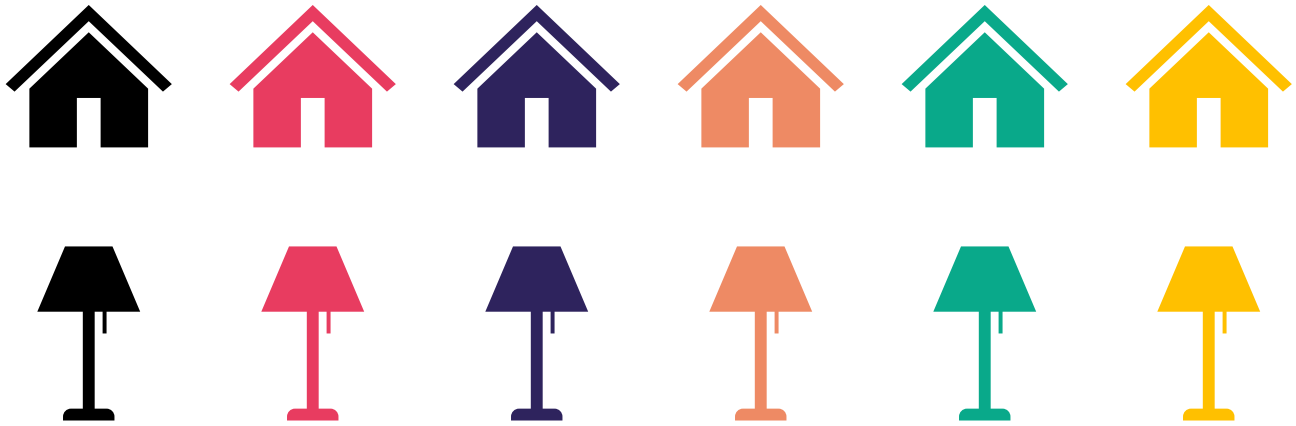
`int[]`

- Not really what an array is for
- Again, just two `int`s – just have to "trust" that we'll remember to treat it like a point

Let's make a class instead!

(PCM) Instance Methods

- Instance methods are defined in a class
- Calling an instance method on a particular *instance* of the class will have effects on *that* instance



(PCM) Instance Methods

- Instance methods are defined in a class
- Calling an instance method on a particular *instance* of the class will have effects on *that* instance

Lecture Outline

- Announcements
- SearchEngine Recap
- Bias in Data Discussion
- OOP Review
- Example
- **Abstraction** 

(PCM) Abstraction

The separation of ideas from details, meaning that we can *use* something without knowing exactly *how* it works.

You could use the `Scanner` class without understanding how it worked internally!

(PCM) Client v. Implementor

We have been the *clients* of many objects this quarter!

Now we will become the *implementors* of our own objects!