UNIVERSITY of VIRGINIA

# CS 1112: Introduction To Programming

## Introduction to Python Classes

Creating our own data types!

Dr. Nada Basit // basit[*at*]Virginia[*dot*]edu

Spring 2024

# Friendly Reminders

- Your ***safety*** and ***comfort*** is important!
  - If you choose to wear a mask you are welcome to do so
  - *We will interpret wearing a mask as being considerate and caring of others in the classroom (not that you are sick), and realize that some may choose to mask to* ***remain distanced***

- Be an ***active*** participant in your learning!
  You're welcome and ***encouraged*** to ask questions during class!

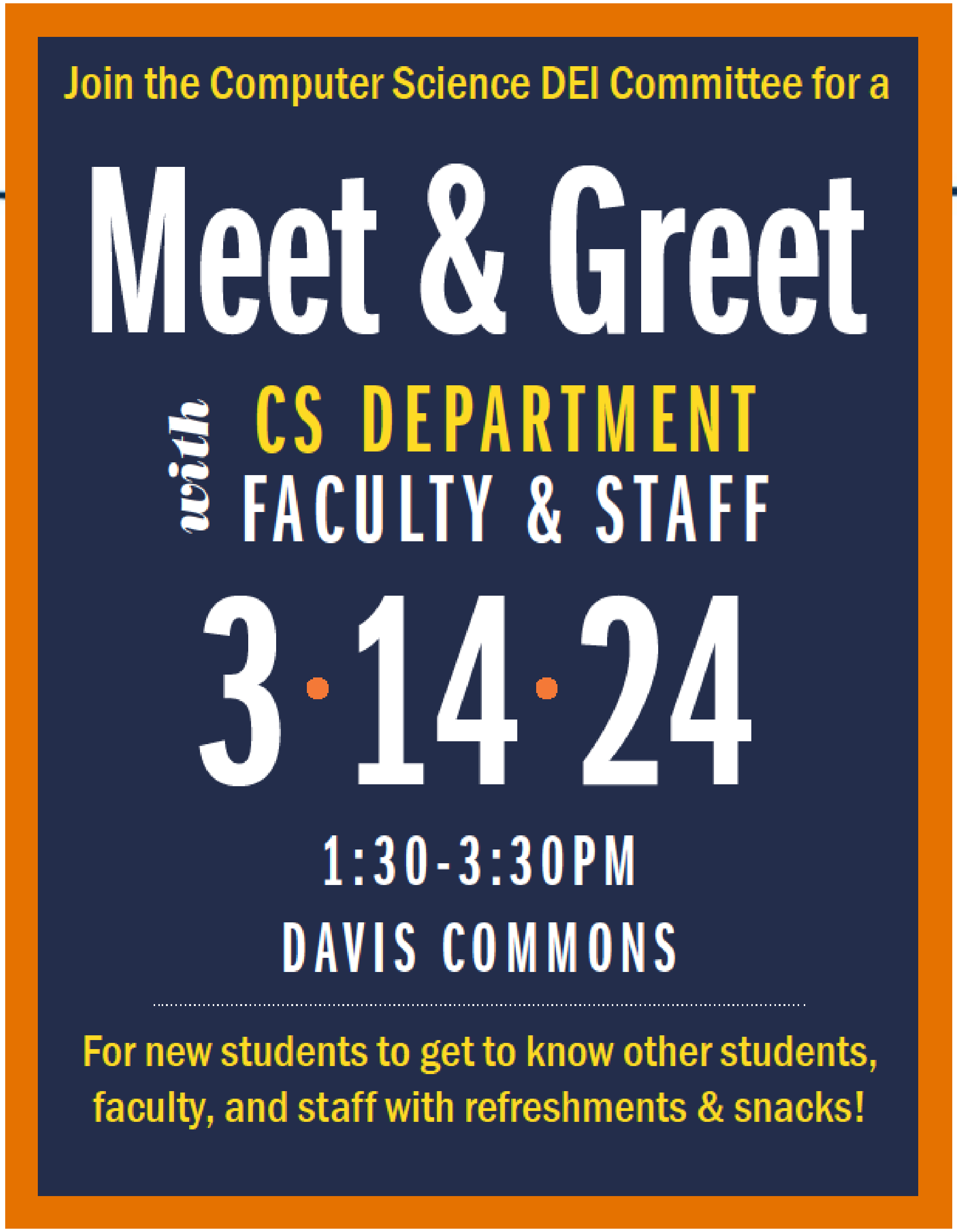
- If you feel ***unwell***, or think you are, please stay home
  - *We will work with you!*
  - Get some rest ☺
  - View the recorded lectures – *please allow 24-48 hours to post*
  - *Contact us!*

## Announcements / Reminders

- Next Quiz will be released on Friday
  - Will be due Monday 3/18
- Next Programming Assignment has been released (yesterday)
  - Will be due Wednesday 3/20

- **Exam 1** error corrections due by Wednesday, March 20 (11:00pm)
  - In-person (Professor or TA)
  - Submitting an .mp3 file (< 10 mins)

## Event!



Join the Computer Science DEI Committee for a

Meet & Greet

*with* CS DEPARTMENT FACULTY & STAFF

3·14·24

1:30-3:30PM

DAVIS COMMONS

For new students to get to know other students, faculty, and staff with refreshments & snacks!

## Classes

A form of encapsulation

A means to create a custom (complex) data type!!

## Classes

- Introduction to structure
- Encapsulation
- Creation of custom (non-native) data types

- Building regions of code larger than functions
  - Functions exist inside classes
  - There are “special” functions that are specific to classes / creation of a data type
- Example on how to use classes and how to create instances of these classes

## Constructor (__init__)

```
class Alien:
    # fields: name, numArms, planet

    def __init__(self, name, numArms, home):  # constructor
        # responsible for creating objects of Aliens (this class)
        # must handle all class attributes (fields)
        self.name = name
        self.numArms = numArms
        self.planet = home
```

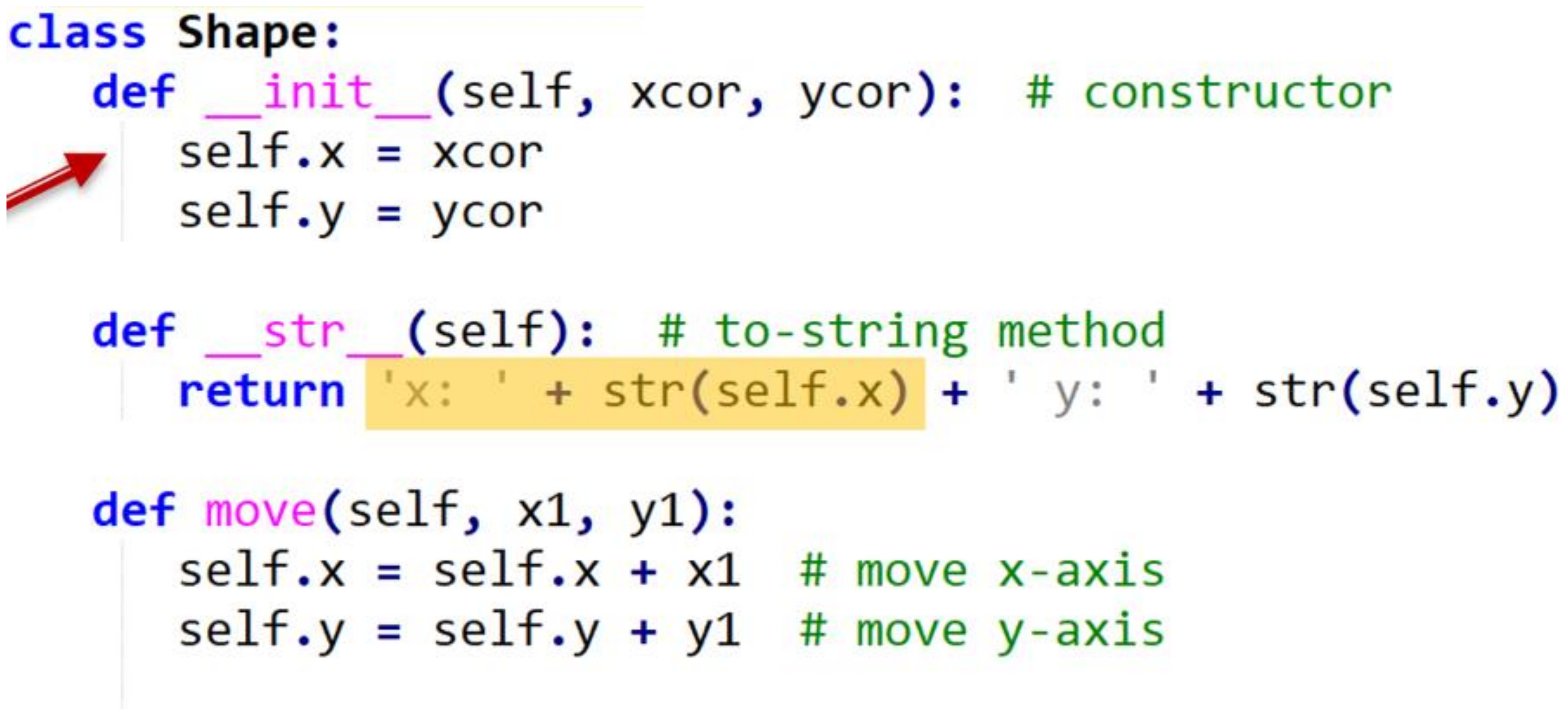
## Printing function (__str__)

```
def __str__(self):
       # to-string method - display the state of the obj
       # (also useful for testing class implementation)
       return self.name + ' ' + self.planet + ' ' + str(self.numArms)


=========================================================================


def function_x(self, a, b, c):
       # Other methods (behaviors) exist within the class
       # All objects of this class exhibit the same behaviors
```

## Example: Shape Class



```
class Shape:
    def __init__(self, xcor, ycor):  # constructor
        self.x = xcor
        self.y = ycor

    def __str__(self):  # to-string method
        return 'x: ' + str(self.x) + ' y: ' + str(self.y)

    def move(self, x1, y1):
        self.x = self.x + x1  # move x-axis
        self.y = self.y + y1  # move y-axis
```

## Let’s Look At Some Examples

- classStructure.py
- classExamples.py
- Alien example (if time)

## Reminder: CS Laptop Loaner Program

- This course requires students to have a **laptop**
- I realize that not everybody might have one (nor necessarily need one for their desired major / path…)
- If you do not have a laptop for any reason… ***not to worry!***
- The CS department's Systems staff has a notebook / laptop loaner program and will be able to loan you a notebook / laptop computer for the duration of the semester if you don't have one or if you cannot afford one.
  - Also available if your laptop is broken and under repair, we can arrange for you to receive a loaner laptop for a week or two until your own laptop is fixed

Interested? Link: https://www.cs.virginia.edu/wiki/doku.php?id=cs_laptop_loaner

*I am happy to be your sponsor. Please let me know.*