

Objectives

- Continuing with OOP

Get out your Graphics API

Feb 1, 2021

Sprenkle - CSCI111

1

1

Review: Object-Oriented Programming

- What is the term for how we create a new object?
 - What is the syntax for that?
- What is the term for how we give commands to/do operations on objects?
 - What is the syntax for that?
- What are two types of methods we talked about?
 - How do they work differently?

Feb 1, 2021

Sprenkle - CSCI111

2

2

Review: Object-Oriented Programming

- How do we create a new object?
 - Using the object's **constructor**
 - The constructor's name is the same as the class name
 - Example:
 - `obj = ClassName()`
 - Creates an object of type `ClassName`
- What is the term for how we give commands to/do operations on objects?
 - **Methods**

Feb 1, 2021

Sprenkle - CSCI111

3

3

Review: Object-Oriented Programming

- What is the syntax for calling a method on an object?
 - `obj.methodName(parameters)`
- What are two types of methods we talked about? How do they work differently?
 - **Accessors** – get information about the object
 - We tend to save that information in a variable, e.g.,
 - `info = obj.methodname(parameters)`
 - **Mutators** – change the state of the object
 - Call the method; don't set it equal to something

Feb 1, 2021

Sprenkle - CSCI111

4

4

Review: What Does This Code Do?

- Use OO terminology previously defined

Need to import the code from graphics.py into our program

```
from graphics import *  
  
win = GraphWin("My Circle", 200, 200)  
point = Point(100, 100)  
c = Circle(point, 10)  
c.draw(win)  
win.getMouse()
```

GraphWin
object
Also known as an
instance of the
GraphWin class

Constructor

Method called on GraphWin object

Note: Class names start with capital letters,
Method names start with lowercase letters

Feb 1, 2021

5

What is the purpose of this line?

- `from graphics import *`
- `graphics.py` – a third-party module, not built into Python
- So that we can use the code from that module in our code, we use
 - `from graphics import *`

Feb 1, 2021

Sprenkle - CSCI111

6

6

Review: Benefits of Object-Oriented Programming

- **Abstraction**
 - Hides details of underlying implementation
 - Easier to change implementation
- Easy reuse of code
 - Can import the library in multiple files
- Collects related data/methods together
 - Easier to reason about data
- Less code in main program
 - Our program code is relatively simple

Feb 1, 2021

Sprenkle - CSCI111

7

7

Programming with the Graphics Library

- Algorithm for our program
 - Create an instance of a 50x100 Rectangle
 - Draw the rectangle
 - Shift the instance of the Rectangle class to the right 10 pixels
 - Display (print) the x- and y- coordinates of the upper-left corner of the Rectangle
- Now, implement it!
 - Draw on paper to help you think it through
 - Refer back to example program

Feb 1, 2021

Sprenkle - CSCI111

[rectangle.py](#)

8

8

Post-mortem: Analyzing Problem-Solving Process

- There were gaps in our algorithm
 - We needed a GraphWin
 - We needed to import graphics.py
 - Algorithm didn't say where to put the rectangle
 - Don't forget to wait for the mouse click and then close
- We didn't necessarily work linearly
 - Iteration often involves working backwards or in circles or ...

Feb 1, 2021

Sprenkle - CSCI111

9

9

Problem: Draw a Full-Canvas Tic-Tac-Toe Board

- Using the Graphics API
- Make lines purple with line width 3
- The width and height of the canvas is 200

Feb 1, 2021

Sprenkle - CSCI111

tictactoe.py

10

10

Modification to Tic-Tac-Toe

- **clone** a vertical line and horizontal line and shift appropriately
- Why clone?
 - Maintain the same properties (color, line-width, length)
 - Simplifies code

tictactoe_clone.py

Feb 1, 2021

Sprenkle - CSCI111

11

11

Review: What is Our Process?

- Import the Graphics Library
- Create the GraphWin
- Construct the Object
 - Construct the objects it needs
 - Set up its color, width, ...
- Draw the object
- Call getMouse and close to make the window stay open

Feb 1, 2021

Sprenkle - CSCI111

12

12

Looking Ahead

- Lab 2 Prep – due before class
 - Basically, end of Chapter 2, beginning of Chapter 4
- Lab 2 Tuesday
 - What are you going to draw?