

# CSC 111:

# Intro to Computer Science through Programming

Spring 2017  
Prof. Sara Mathieson

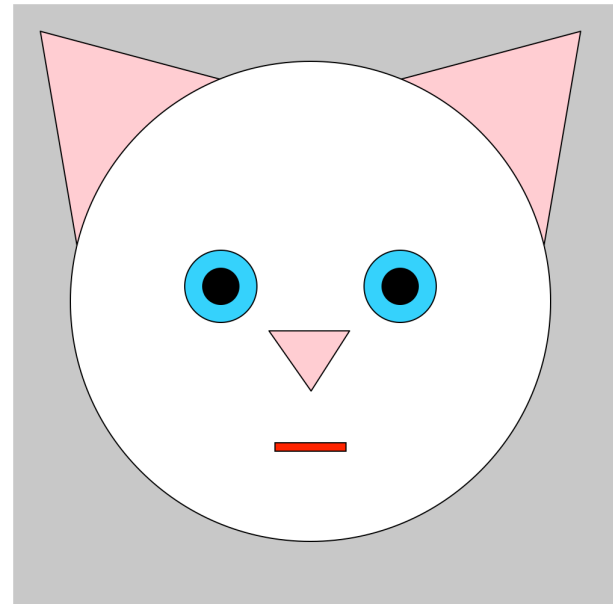


# Admin

- + Homework 5 due Tues after spring break
- + Transcripts from now on: only successful output at the end

# Outline: 3/10

- + Recap last time
- + Continue OOP
- + Continue Graphics
- + Cats in Graphics



Cat by Joe O'Rourke

Recap

# Informal quiz (discuss with a partner)

- 1) **moon** is an \_\_\_\_\_ of the **Circle** \_\_\_\_\_
- 2) **Circle(..., ...)** is a \_\_\_\_\_
- 3) **Point(200, 200)** and **100** are \_\_\_\_\_
- 4) **setFill(...)** and **draw(...)** are \_\_\_\_\_ not \_\_\_\_\_

```
moon = Circle(Point(200, 200), 100)
moon.setFill("orange")
moon.draw(win)
```

# Informal quiz (discuss with a partner)

- 1) **moon** is an *instance* of the **Circle** class
- 2) **Circle(..., ...)** is a \_\_\_\_\_
- 3) **Point(200, 200)** and **100** are \_\_\_\_\_
- 4) **setFill(...)** and **draw(...)** are \_\_\_\_\_ not \_\_\_\_\_

```
moon = Circle(Point(200, 200), 100)
moon.setFill("orange")
moon.draw(win)
```

# Informal quiz (discuss with a partner)

- 1) **moon** is an *instance* of the **Circle** class
- 2) **Circle(..., ...)** is a *constructor*
- 3) **Point(200, 200)** and **100** are \_\_\_\_\_
- 4) **setFill(...)** and **draw(...)** are \_\_\_\_\_ not \_\_\_\_\_

```
moon = Circle(Point(200, 200), 100)
moon.setFill("orange")
moon.draw(win)
```

# Informal quiz (discuss with a partner)

- 1) **moon** is an *instance* of the **Circle** class
- 2) **Circle(..., ...)** is a *constructor*
- 3) **Point(200, 200)** and **100** are *parameters*
- 4) **setFill(...)** and **draw(...)** are \_\_\_\_\_ not \_\_\_\_\_

```
moon = Circle(Point(200,200), 100)
moon.setFill("orange")
moon.draw(win)
```



# Informal quiz (discuss with a partner)

- 1) **moon** is an *instance* of the **Circle** class
- 2) **Circle(..., ...)** is a *constructor*
- 3) **Point(200, 200)** and **100** are *parameters*
- 4) **setFill(...)** and **draw(...)** are *methods* not *functions*

```
moon = Circle(Point(200,200), 100)
moon.setFill("orange")
moon.draw(win)
```

Continue: OOP

# Instance Variables

- + We have already seen example of methods, which belong to a specific instance of a class. Example: `<Point instance>.draw(...)`

```
>>> p = Point(50,75)
>>> p.draw(win)
```

# Instance Variables

- + We have already seen example of methods, which belong to a specific instance of a class. Example: `<Point instance>.draw(...)`

```
>>> p = Point(50,75)
>>> p.draw(win)
```

- + Instances also have *instance variables*, which can be accessed and/or modified. We also use a “dot”, but no parentheses!

```
>>> p.x
50.0
>>> p.y
75.0
>>>
```

# GraphWin class

- + **GraphWin(title, width, height)** – constructs a new graphics window (default width and height are both 200)
- + **setBackground(color)** – set the background color
- + **close()** – closes the window
- + **getMouse()** – waits for the user to click, returns the click position as a **Point**

# Methods for all Graphics Objects

- + **setFill(color)** – sets the interior color of an object
- + **setOutline(color)** – sets the outline color of an object
- + **setWidth(pixels)** – sets the outline width (doesn't work for **Point**)
- + **draw(window)** – draws the object on the given window
- + **undraw()** – removes the object from a graphics window
- + **move(dx,dy)** – moves the object dx in the x direction and dy in the y direction
- + **clone()** – returns a duplicate (new copy) of the object

# Point class

- + **Point(x,y)** – constructs a new point at the given position
- + **getX()** – returns the current x coordinate
- + **getY()** – returns the current y coordinate

# Line class

- + **Line(point1, point2)** – constructs a line from point1 to point2
- + **setArrow(string)** – sets the arrowhead of a line (“first”, “last”, “both”, “none”)
- + **getCenter()** – returns the midpoint of the line
- + **getP1(), getP2()** – returns a clone of the corresponding endpoint



# Circle class

- + **Circle(center, radius)** – constructs a circle at the given position and with the given radius
- + **getCenter()** – returns a clone of the center point
- + **getRadius()** – returns the radius
- + **getP1(), getP2()** – returns a clone of the corresponding corner of the circle's bounding box (upper left, lower right)

# Rectangle class

- + **Rectangle(point1, point2)** – constructs a rectangle with opposite corners at the given points (upper left, lower right)
- + **getCenter()** – returns the center point
- + **getP1(), getP2()** – returns a clone of the corner point

# Polygon class

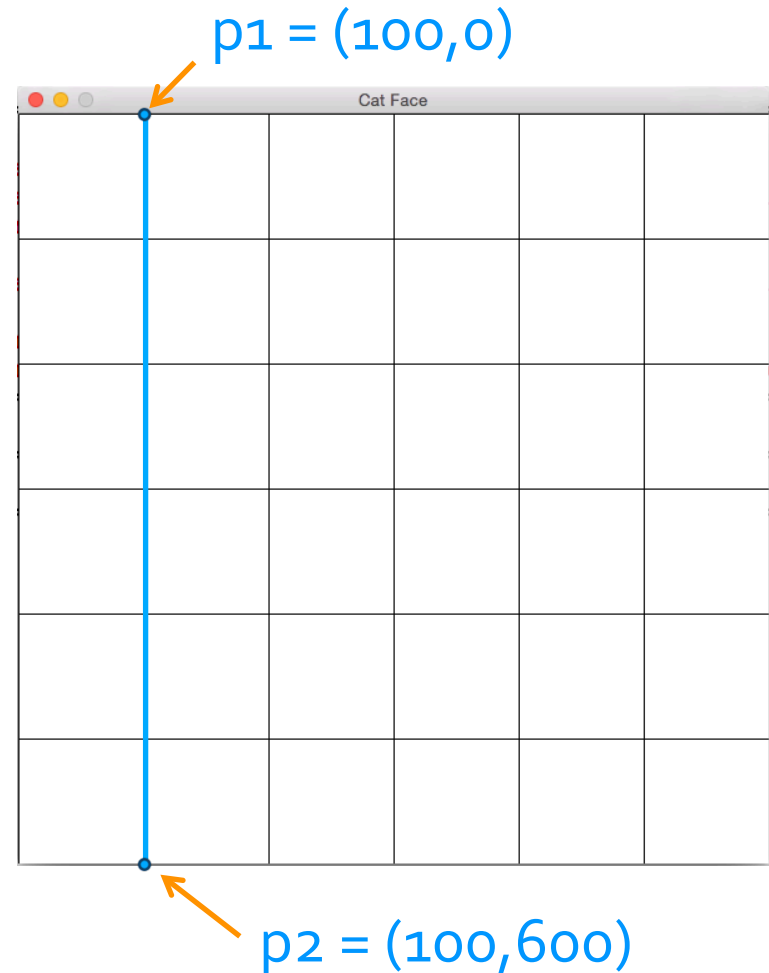
- + **Polygon(point1, point2, point3, ...)** – constructs a polygon with the given points as vertices (also accepts a list of points)
- + **getPoints()** – returns a list of the points in the polygon

# Cat Face Exercise

# Step 1: create a grid

- + Window 600 x 600
- + Grid lines every 100
- + Line example:

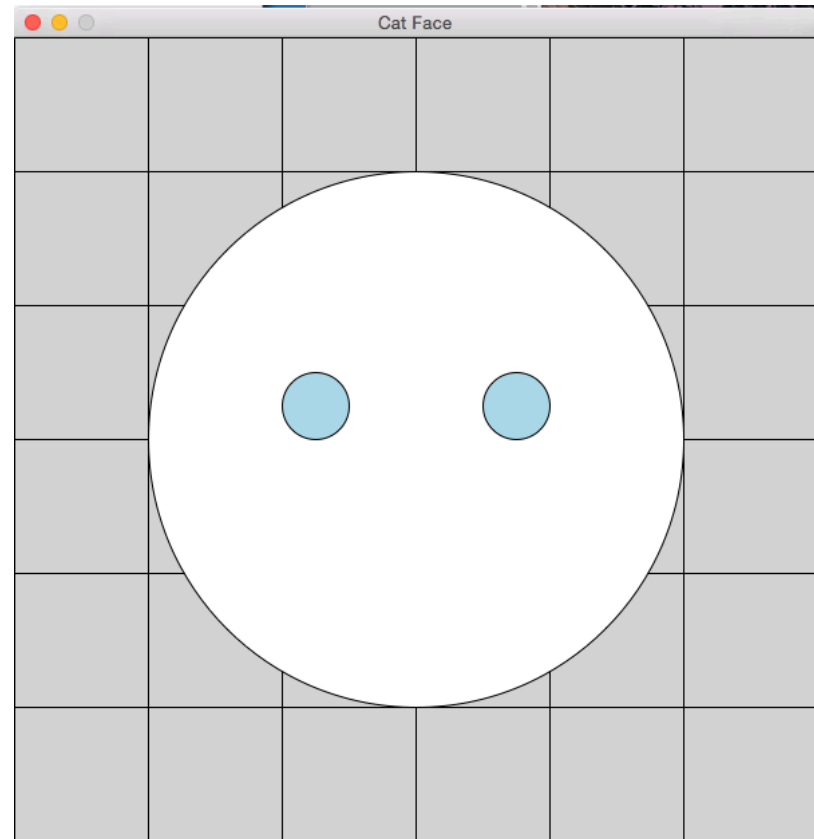
```
# first vertical line  
p1 = Point(100,0)  
p2 = Point(100,height)  
l = Line(p1,p2)  
l.draw(win)
```



## Step 2: create a face and eyes

- + Create a left eye using a circle
- + Clone (copy) the left eye to make the right eye
- + Move the right eye over

```
right_eye = left_eye.clone()  
right_eye.move(dx,dy)  
right_eye.draw(win)
```



## Step 3: create nose, ears, mouth

- + Create mouth as a rectangle
- + Create nose as a polygon
- + Create ears as polygons
- + Remove background grid
- + Change colors!

<http://wiki.tcl.tk/37701>

