creativity & computation lab

week 3 || object oriented programming

review

WHERE WE HAVE BEEN

What we have done:

- Presentations!!
- Transformations
    - translate()
    - pushMatrix(); popMatrix();
- Trigonometry
    - angles/radians
    - coordinate systems (Cartesian/Polar)
    - sin()
    - cos()
    - oscillation

agenda

WHERE WE ARE GOING

What’s on for today:

- Citing code and authorship
  - About time, right?
- Object-oriented programming!
  - What we’ve all been waiting for...

first things later (oopsies.)

HOW TO CITE YOUR CODE!

There is no standard way to cite authorship and source code that you have used. We will use this model:

```
/*
 * Title of the sketch
 * Author
 * Date
 * Short Description of what it does
 * Code adapted from [title of piece]
 * by [name]. Source code can be found
 * here:_________.
 */
```

object oriented programming

LIFE CHANGING

Before you use OOP

object oriented programming

LIFE CHANGING

After you know OOP

object oriented programming

LIFE CHANGING

Revo + lution/lation

Includes everything you have already learned.
//variables, functions, conditionals, loops, etc.

It is simply a different way of structuring them in your code.
//more modular, more intuitive, more reusable

OOP allows you to take ALL of the variables and functions out of the main program and consolidate them into an object.

This means you only have one variable (the object) instead of a zillion!

warm up!

FOOD: THE WAY TO A STUDENT'S BRAIN

Everyone loves mini cupcakes. Period.

1) You will each receive a cupcake. Do not eat it! (Yet.)

2) What are the properties and actions associated with a cupcake?

3) Write out the following pseudocode for the cupcake you have. Include its properties (data) and actions (functions).

warm up!

PSEUDOCODE

Peanut Butter Cup. Yum.

```
//Properties = data
int cupcakeSize;
boolean filling;
color icing;

//Actions = methods (aka functions, but called methods here)
void bake();
void fill();
void ice();
void stuffYourFace();
```

warm up!

FOOD: THE WAY TO A STUDENT'S BRAIN

Time to code!

4) Exchange pseudocode with a person sitting near you.

5) Create a sketch of their cupcake.

object oriented programming

LIFE CHANGING

Let’s look at our cupcakes.

Let’s say you wanted to create a sketch that had 10 cupcakes - of all different kinds! Crazy talk! (And maybe a stomach ache)

Right now, knowing what we know, we would have lots of variables, probably some arrays, a few loops, and a dash of conditionals.

While this doesn’t sound too hard, it is confusing to look at and NOT the most efficient way to create many of the same thing.

warm up!

FOOD: THE WAY TO A STUDENT'S BRAIN

Instead, we can create a template or class (cupcake tin if you will) that will allow us to create lots of cupcakes objects easily.

Welcome to object-oriented programming!

Class

//template

Object

//cupcake

object oriented programming

OBJECTS AND CLASSES

OBJECT

Data structures that consist of data fields (properties) and methods (actions/functions).

Objects are instances of a class.

This means that you can use your template to easily create lots of objects that all have DIFFERENT properties.

CLASS

A template for creating objects.

RE: a description of the object's properties and actions.

object oriented programming

OBJECTS AND CLASSES

OBJECT

Data structures that consist of data fields (properties) and methods (actions/functions).

Objects are instances of a class.

This means that you can use your template to easily create lots of objects that all have DIFFERENT properties.

CLASS

A template for creating objects.

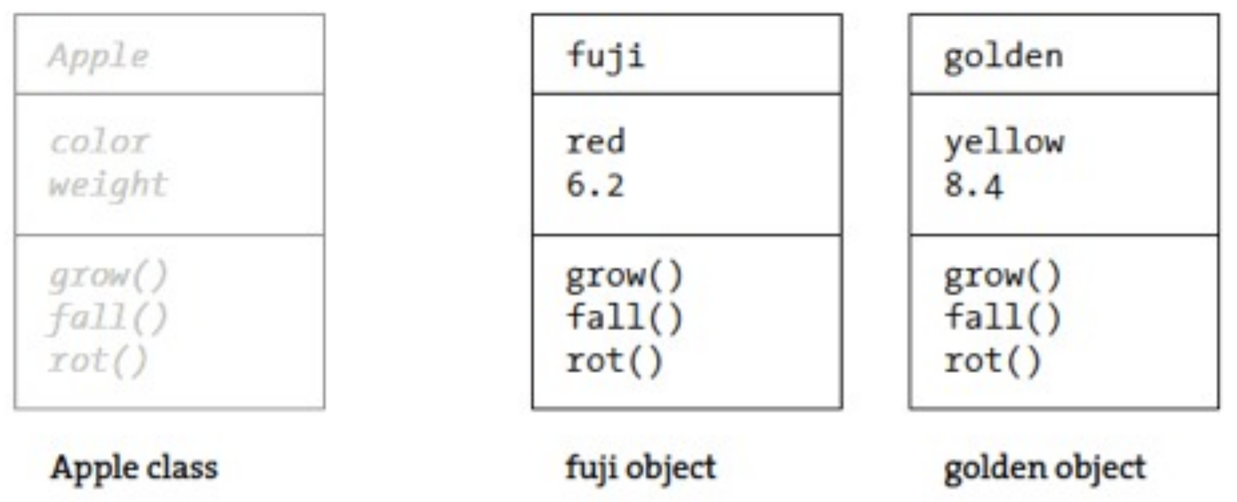
RE: a description of the object's properties and actions.

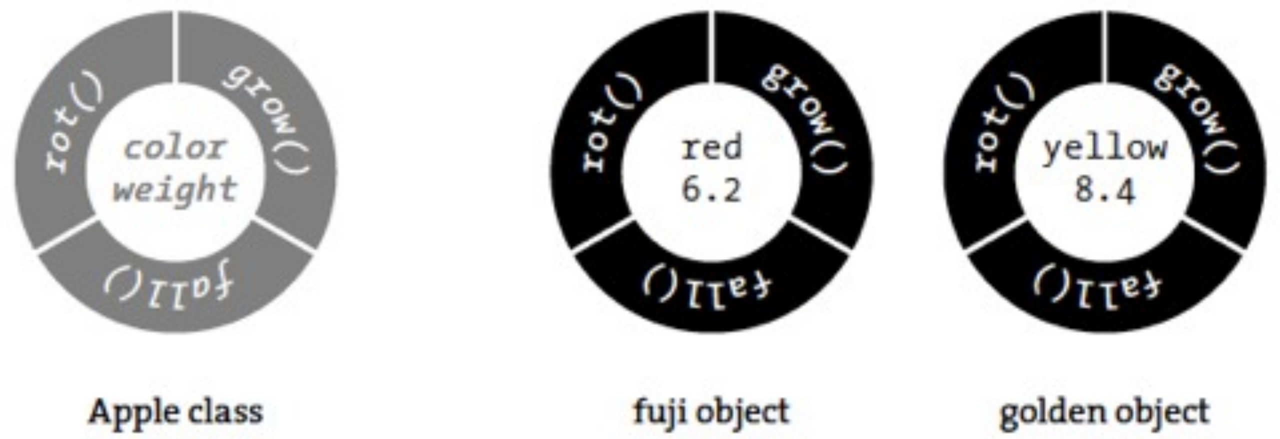
Let's let this sink in for a sec....

object oriented programming

OBJECTS AND CLASSES

Let's think of some other examples...





object oriented programming

CLASS FILE

```
class Cupcake{
    int cupcakeSize;
    boolean filling;
    color icing;

Cupcake(){
    cupcakeSize = huge;
    filling = true;
    icing = color(chocolate);
}

void bake(){
//code here
}
void fill(){
//code here
}
void ice(){
//code here
}
void stuffYourFace(){
//code here
}
}
```

CLASS FILE: There are four parts:

Class name

//Name it anything you want, but name it well!

Data

//The properties of the class - what makes it special!

Constructor

//You must actually *construct* the object in the class - this will allow you to construct instances of it in the main file

Functionality

//Give it something to do!

object oriented programming

CLASS FILE

```
class Cupcake{
    int cupcakeSize;
    boolean filling;
    color icing;

Cupcake(int newCupcakeSize){
    cupcakeSize = newCupcakeSize;
}

void bake(){
//code here
}
void fill(){
//code here
}
void ice(){
//code here
}
void stuffYourFace(){
//code here
}
}
```

Constructor

Why do we create a new variable as an argument to pass?

Because we don't want to be able to access it outside of the class.

RE: This is best programming practice.

object oriented programming

MAIN FILE

In the MAIN FILE you MUST do three things:

```
Cupcake peanutButter;

void setup(){
peanutButter = new Cupcake();
}

void draw(){
peanutButter.cupcakeSize = small;

peanutButter.bake();
peanutButter.fill();
peanutButter.ice();
peanutButter.stuffYourFace();
}
```

Declare

//Name it anything you want, but name it well!

Initialize

//You must actually *construct* the object in the class - this will allow you to construct instances of it in the main file

Use

//Call methods and assign new variables

object oriented programming

PUTTING IT ALL TOGETHER

CLASS FILE

```
class Cupcake{  //Class name
//Class data - properties of the class
    int cupcakeSize;
    boolean filling;
    color icing;

Cupcake(){ //Constructor
    cupcakeSize = huge;
    filling = true;
    icing = color(chocolate);
}
//Class methods
void bake(){
//code here
}
void fill(){
}
void ice(){
}
void stuffYourFace(){
}
}
```

MAIN FILE

```
Cupcake peanutButter;
//Declare your object

void setup(){
//Instantiate each new object
peanutButter = new Cupcake();
}

void draw(){
//Call methods
peanutButter.cupcakeSize = small;

peanutButter.bake();
peanutButter.fill();
peanutButter.ice();
peanutButter.stuffYourFace();
}
```

object oriented programming

YOU KNOW WHAT'S COMING

So what do you think is next
on the agenda?

object oriented programming

THE 3 STEPS YOU MUST TAKE TO CREATE AN OBJECT

	Primitive data type //we know this stuff	Complex data type //eh, not just yet
1. DECLARE	`int xPos;`	`Cupcake peanutButter;`
2. INITIALIZE/ASSIGN	`xPos = 100;`	`peanutButter = new Cupcake;`
3. USE	`rect(xPos,50,20,20);`	`peanutButter.stuffYourFace;`

object oriented programming

MORE BOUNCING BALLS

Let’s take a look at another example...

object oriented programming

YOU KNOW WHAT'S COMING

So what do you think is next
on the agenda?

IN CLASS EXERCISE!

object oriented programming

YOU KNOW WHAT'S COMING

Turn your cupcake into an object!

Make a class using the code you created at the being of class.

Make at least 3 objects of your cupcake that are somehow different.

$\rightarrow$

$\rightarrow$