OpenFrameworks

1. cd (copy and past of folder)

2. mkdir NAMEFOLDER

3. ls // checks what's int he folder

4. cd NAMEFOLDER

5. touch README.md

// push everything to git

1. cd repository

2. ls

3. git add . //staging everything

4. git commit -m "MESSAGE"

5. git push -u origin master

Cartesian Systems

Variables:

Void setup {

// runs once

}

void draw {

// loops

}

1. All variables must have a name and a datatype

Data Types

1. int : whole numbers
2. **float:**
3. **string**
4. Boolean
5. Color

- You cant add different data types

Structure:

1. Place declarations in void setup {}

Debugging:

cout << “Hello World!” << endl

Class Code:

void testApp::diamond(ofPoint center, float width, float height){

//line 1

ofLine(center.x - width/2, center.y, center.x, center.y-height/2);

// line 2

ofLine(center.x, center.y-height/2, center.x+width/2, center.y);

//line 3

ofLine(center.x+width/2, center.y, center.x, center.y+height/2);

//line 4

ofLine(center.x, center.y + height/2, center.x - width/2, center.y);

}

1. cd (copy and past of folder)

2. mkdir NAMEFOLDER

3. ls // checks what's int he folder

4. cd NAMEFOLDER

5. touch README.md

// push everything to git

1. cd repository

2. ls

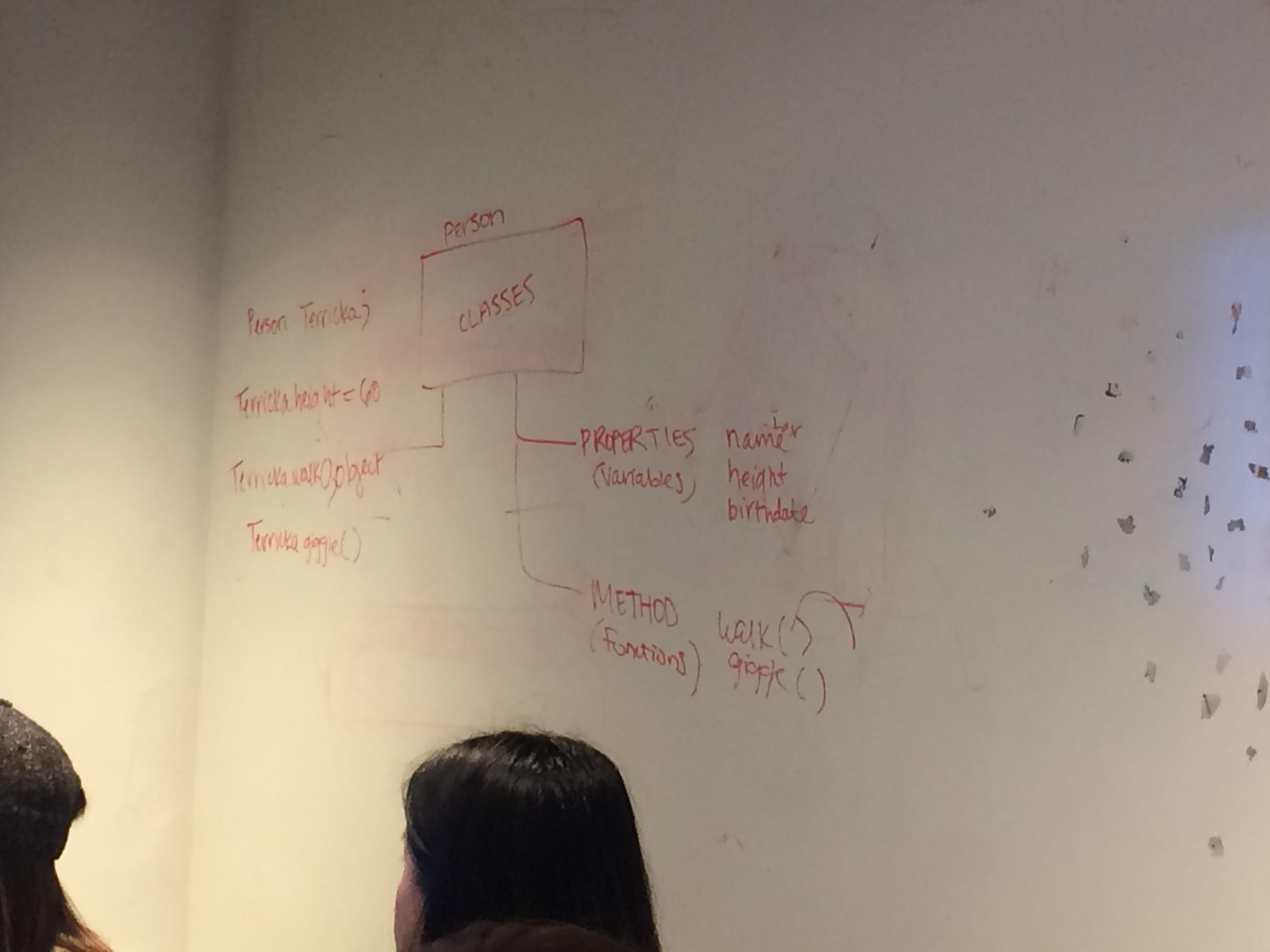
3. git add . //staging everything

4. git commit -m "MESSAGE"

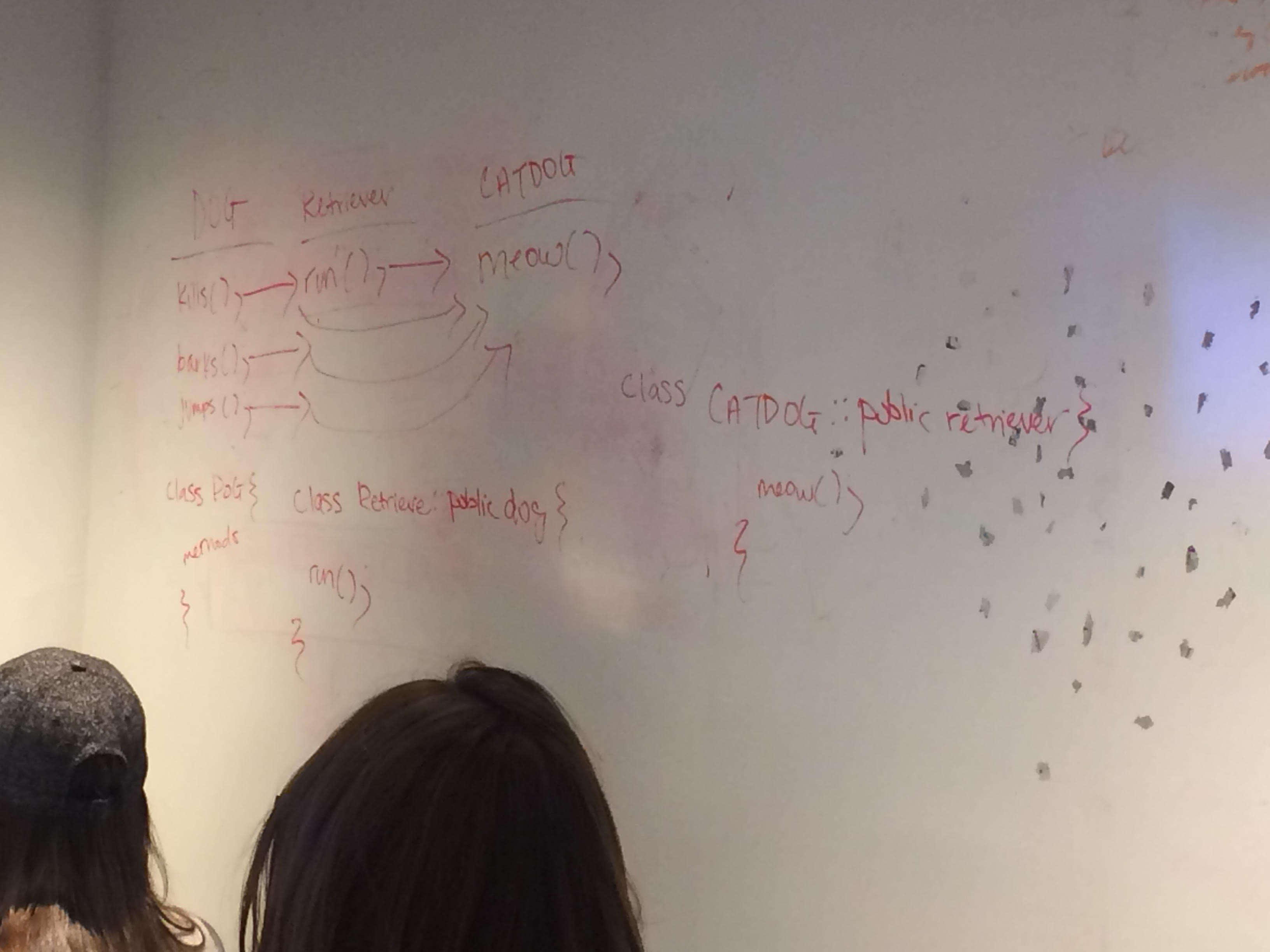
5. git push -u origin master

ofLine(center.x - width/2, center.y, center.x, center.y-height/2);

Classes and Objects



Inheritance, public and private classes



class Dog{

//properties

int age;

color fur;

int size;

string breed;

//constructor

Dog(){

age = 1;

size = 10;

}

//methods

void bark();

void jump();

void kill();

}

//new instance

Dog chester = new Dog();