Girls Who Code, Week 8

Object Oriented Programming

WIT Shout-Out of the Week: Jacki Morie

- One of the first prominent women working in Virtual Reality
- She earned a bachelor's degree in Fine Arts from Florida Atlantic University and then two masters degrees in Fine Arts and Computer Science from the University of Florida, and a PHd in Immersive Environments from University of East London
- Worked developing/researching Virtual Reality softwares at University of Central Florida, US Army, Disney
- Helped with the creation of multi-sensory environments and pioneered medical use for virtual reality

sdk photo

CYBERFACE 2
LEEP BOSTON
LEEP SYSTEMS, INC

Videos

https://www.youtube.com/watch?v=nrcj-90M-f8

https://www.vice.com/en_us/article/8qx7dx/this-afterlife-experience-is-everything-thats-wrong-with-VR-hype

Warm Up Activity

Recursion Refresher

- Create a program that calculates an exponential expression
- You will need a base and an exponent
- What is the base case?
- What is the recursive case?

Ex:

$2^4 = 16$; base = 2, exponent = 4, result = 16

Object-Oriented Programming

Object Oriented Software Design

- In programming it is helpful to compartmentalize (or group together) a set of functions, data, or attributes. This practice is helpful when testing/debugging code, you can find out which sections are faulty this way.

- This style of coding is called Object Oriented Programming: a style of programming that allows you to model real world concepts in order to create complex programs.

Students

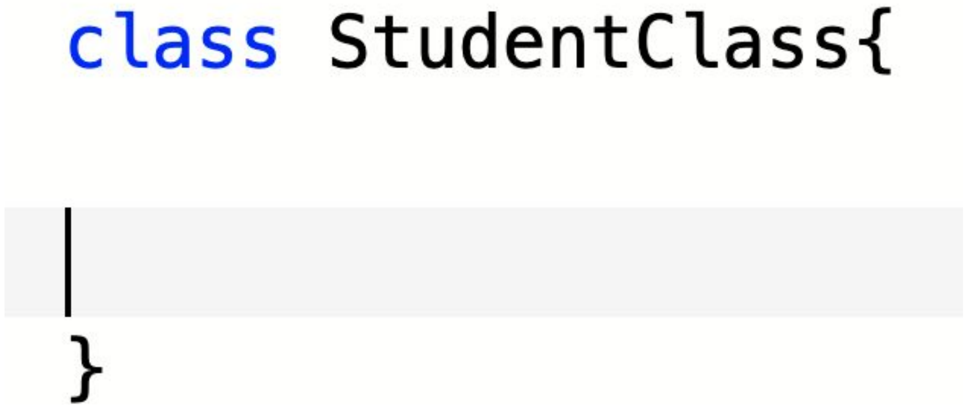
Class Definition

- A class is a grouping of code that is used to represent an ***object***
    - A class contains methods and attributes that apply to a specific object
    - Classes are ways to model objects in code that is both reusable but also distinct in nature

- Thus far in C# we have always written code in the “MainClass”, this is because in C# all code must be contained within a class and this is the class we have defined that will be our “driver program” and will contain the ***main method*** as we described in our methods lesson

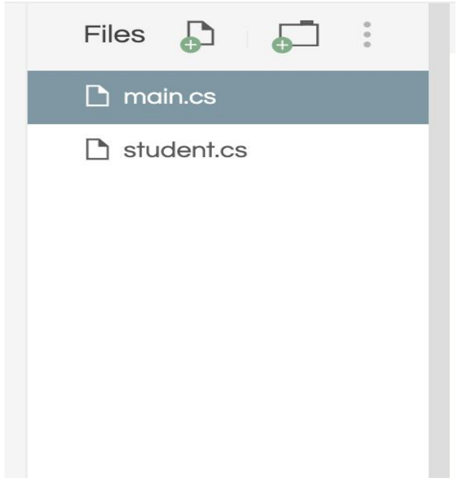
Classes

- Create a brand new file that is named the same as you will name your class
  - Each class gets its own file
- Start with the class keyword
- Then put the name of your class -- good to use camel case



```
class StudentClass{

}
```



Click the file tab to make a new file and name it student.cs to create your student class

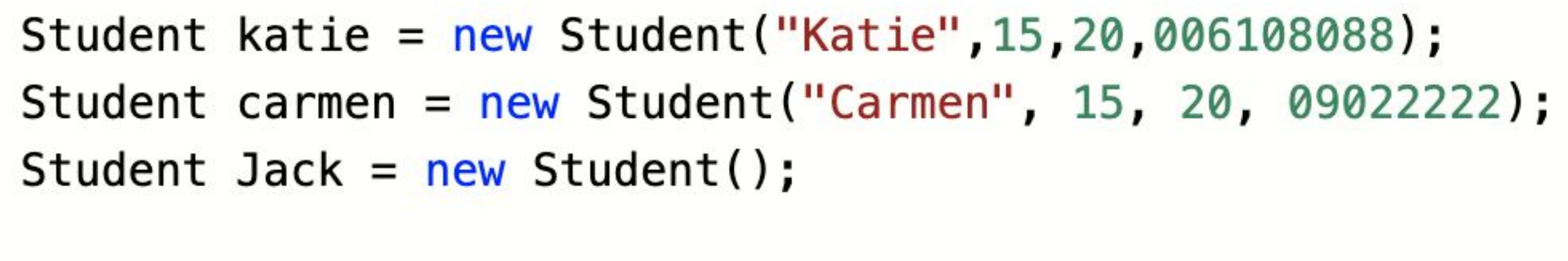
Objects Definition

- **An Object is an Instance of a Class**
  - An object is an entity or item that is one piece of a whole class of items
  - A single student -- Katie, is one student object that contains all the attributes defined in the Student Class

- <u>Instance:</u> a single moment or illustration of an item or event that is one piece of a larger overall picture

- Objects are able to access all the code written in their class

Object examples



```
Student katie = new Student("Katie",15,20,006108088);
Student carmen = new Student("Carmen", 15, 20, 09022222);
Student Jack = new Student();
```

How to create an object:

NameOfClass object-variable = new NameOfClass(attributes);

Attributes

- If we were to model our student class we must first consider what the shared similarities of all general students must be
  - These are called <u>attributes:</u> an attribute is a shared value that would apply to each object of a class

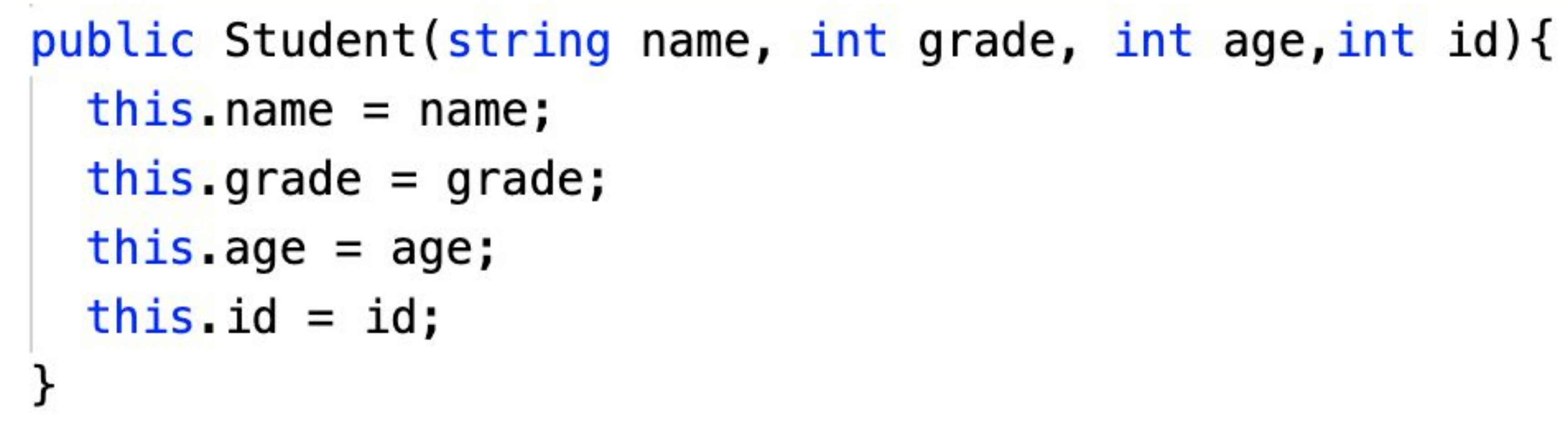
- For a student class we can use any amount of attributes we would like and per each class you create the attributes are changed to fit your programming needs

Constructors

- Constructors are methods that are called immediately when an object is created from the class (during runtime)– it defines what attributes an object has.
- Constructors can have 0 or more parameters -- it simply depends on the object you want to create
    - Sometimes we use constructors with 0 attributes (parameters) and use it as a “default constructor”

Constructor Syntax

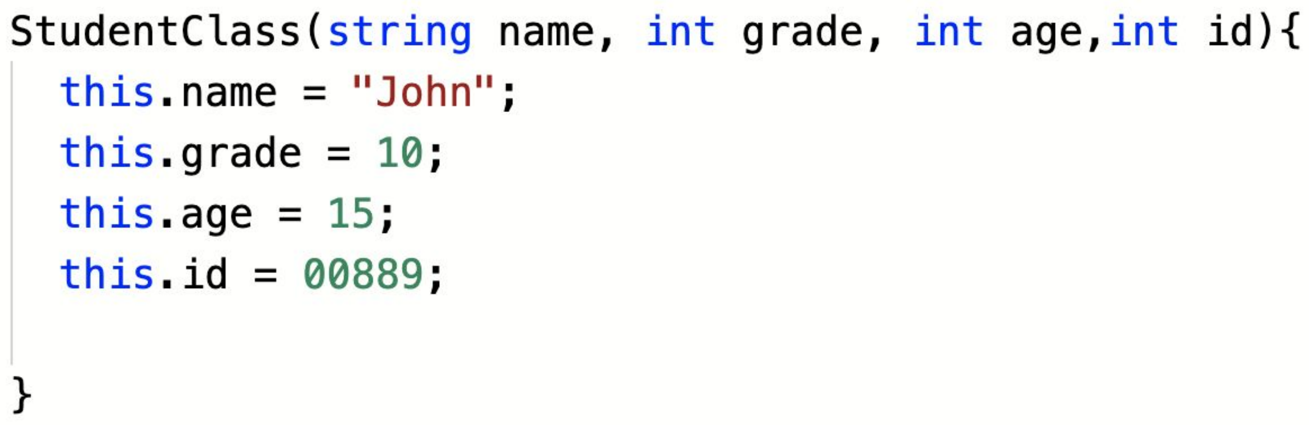
- You need the access modifier, the name of the Class and your paramerters
- The parameters are the attributes of your object
- Assign the parameters with keyword this



```
public Student(string name, int grade, int age,int id){
  this.name = name;
  this.grade = grade;
  this.age = age;
  this.id = id;
}
```

Constructors Continued

- We can also hardcode attributes like such



```
StudentClass(string name, int grade, int age,int id){
  this.name = "John";
  this.grade = 10;
  this.age = 15;
  this.id = 00889;

}
```

- This: keyword refers to the current instance of the class -- an object refers to itself and distinguishes from other class variables with the same name

Multiple (Overloaded) Constructors

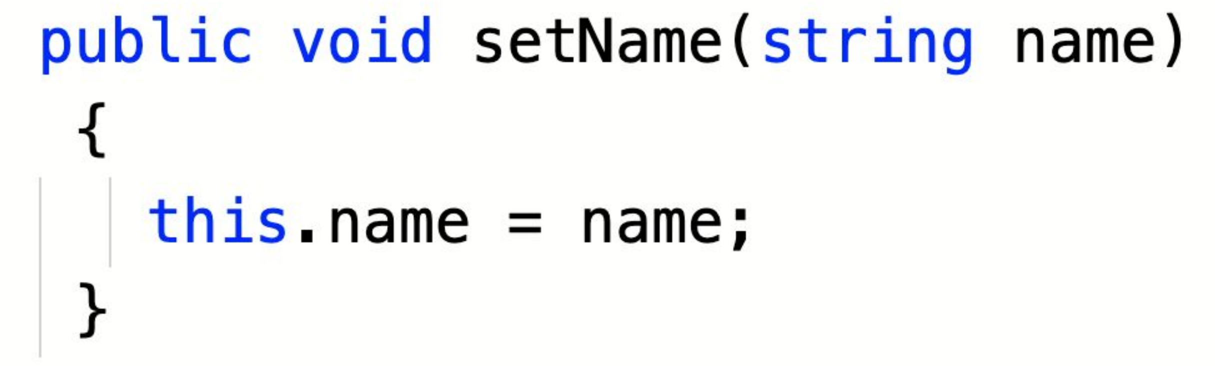
You can have multiple constructors as long as the parameters are different

```
public Student(){
   name = "Student";
   grade = "freshman";
   gender = "female";
   age = 15;
   id = "000000";
 }
```

```
public Student(string name,
string grade, string gender,
string id, int age){
   this.name = name;
   this.grade = grade;
   this.gender = gender;
   this.id = id;
   this.age = age;
 }
```

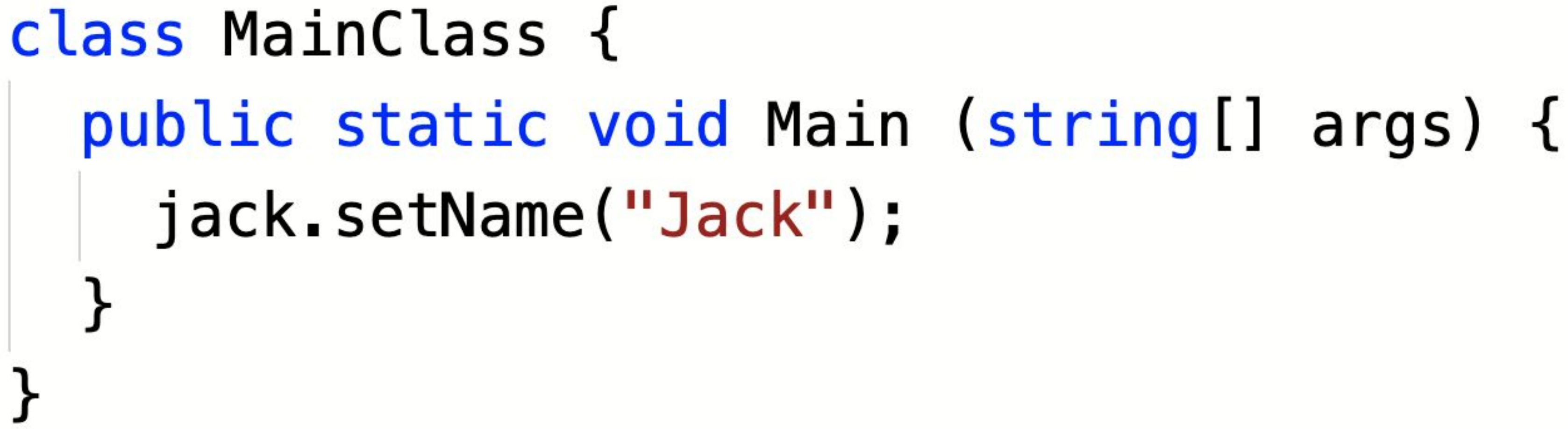
Set Methods (Setters)

- If you create an object and want to change an attribute later, you do so using "setter" methods
  - Setters are methods made within the object class they take a parameter and assign that parameter to the corresponding attribute



```
public void setName(string name)
 {
    this.name = name;
 }
```

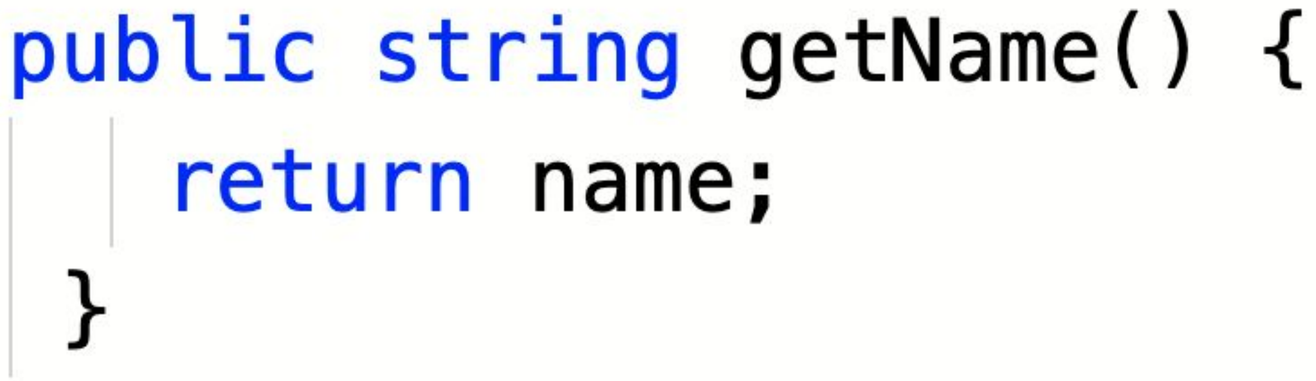
How to set an attribute with the setter method



```
class MainClass {
  public static void Main (string[] args) {
    jack.setName("Jack");
  }
}
```

Get Methods (Getters)

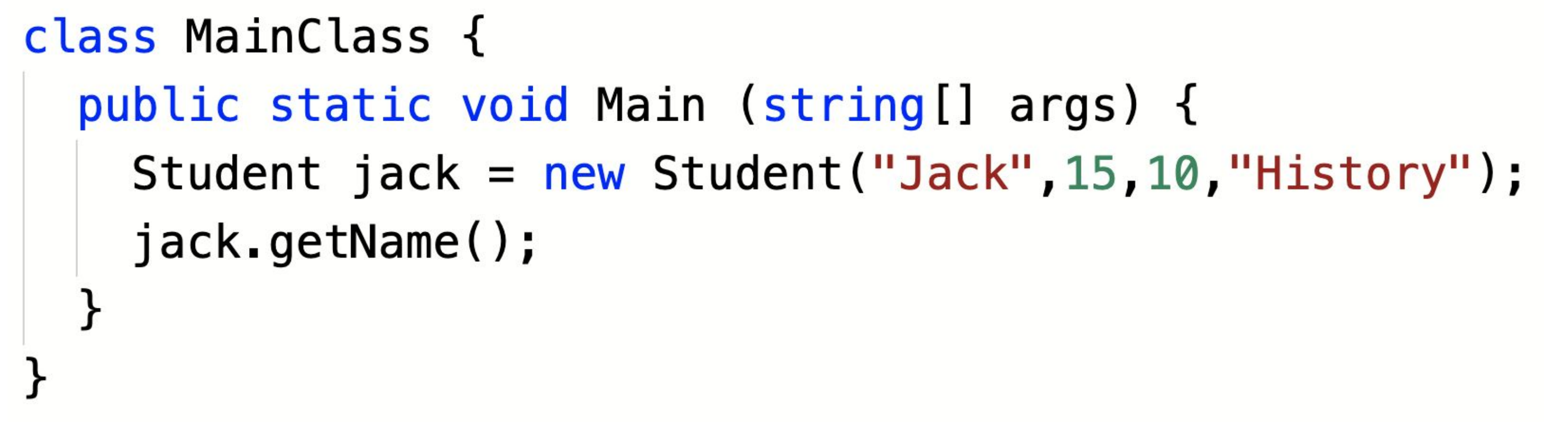
- If you create an object and want to display an attribute, you do so using “Getter” methods
    - Getters are methods made within the object class and return an attribute



```
public string getName() {
    return name;
}
```

How to call the getter method to return an attribute

When you use a getter in your main(), you need to make sure your object is declared



```
class MainClass {
  public static void Main (string[] args) {
    Student jack = new Student("Jack",15,10,"History");
    jack.getName();
  }
}
```

ToString()

You can also create a method for your object that prints all the attributes. Typically, this is done by **overriding** an existing method known as ToString();

```
public override string ToString(){
  //tostring method will print the object
  return $"{name} is {age} years old in grade {grade} and have
  favorite subject {sub} and are allergic to {al}";
}
```

Or:

```
public override string ToString(){
  return "Student "+ this.name + " is in grade " + this.grade;
}
```

Lets Create a Student Class!

Practice: Create a Teacher class

- Create a teacher class that should include the last name of the teacher, the subject they teach, and their age.
- Make sure you add getters and setters
- Make sure you add a ToString override method so that you can print out the object