OOP and Classes

Day 9

Review!
(We aren’t done with this yet)

Procedural Programming

- Coding paradigm in which the problem is broken down into well-defined procedures or steps
- What we were doing till now
- Think about the action first!

Object Oriented Programming

- Coding paradigm in which data plays the main role
- Think about data first!

Problem: Humans want to go to the Moon

Procedural Approach

1. Get human
2. Build spaceship
3. Test spaceship
4. Fly spaceship
5. Land spaceship on moon

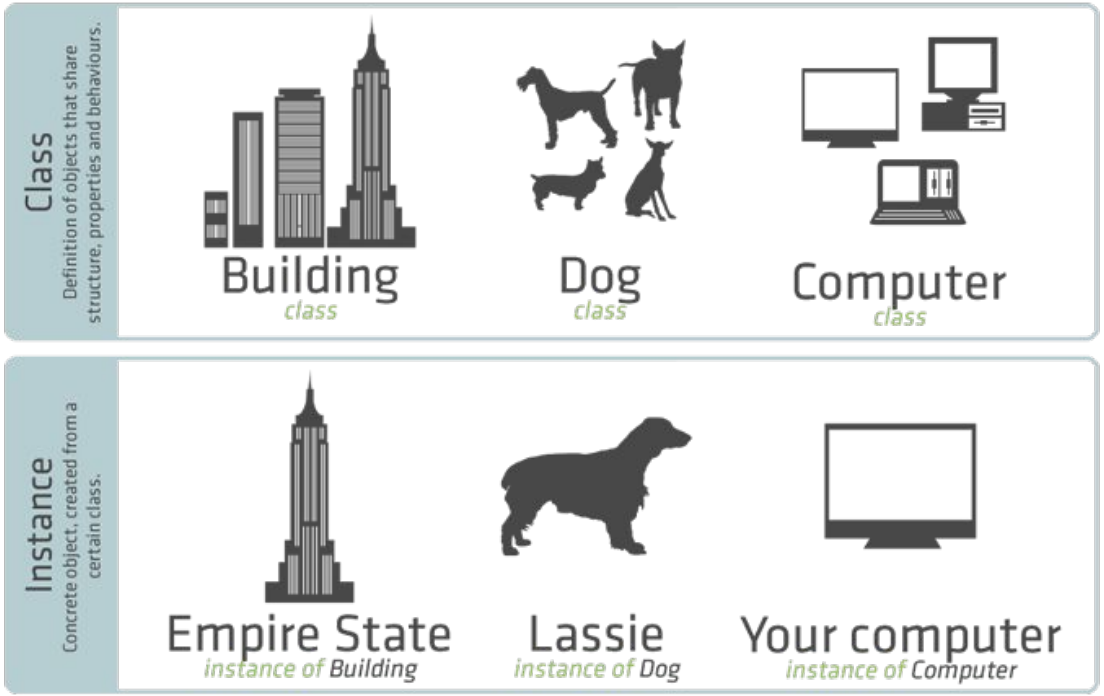
Object Oriented Approach

1. Data types - Human, Moon, Spaceship
2. Properties of spaceship - size, material, power, max no of humans
3. Operations - get, build, test, fly, land

Why OOP?

1. Reusability of code
2. Modularity - easy comprehension
3. Easy to maintain & modify existing code
4. Data hiding for code security

Classes & Objects



How to define a class?

1. Name
2. Data
3. Methods
4. Constructor

How to define a class?

class _____ {

}

How to define a class - Name

```
class superHero {




}
```

How to define a class - Data or Variables

```
class Superhero {

        String realName;
        String power;
        Color suit;


}
```

How to define a class - Methods

```
class Superhero {

        String realName;     //variables
        String power;
        Color suit;

        void protectPeople() {-------}
        int fightBadGuys() {-------}
        void disguise() {-------}
}
```

How to define a class - Constructor

```
class Superhero {

    String realName; //variables
    String power;
    Color suit;

    //methods or functions
    void protectPeople() {-------}
    int fightBadGuys() {-------}
    void disguise() {-------}

    Superhero(){
        realName = “Ted”
        power = “invisibility”
        suit = blue;

    }
}
```

How to define a class - Constructor (Type 2)

```
class Superhero {

    String realName; //variables
    String power;
    Color suit;

    //methods or functions
    void protectPeople() {-------}
    int fightBadGuys() {-------}
    void disguise() {-------}

    Superhero(String realNameX,
    String powerX, Color suitX){

        realName = realNameX;
        power = powerX;
        suit = suitX;

    }

}
```

How to define an object?

```
Superhero superman; //declare the object
Superhero flash;

//initialize
superman = new Superhero (“Clark Kent”, “superhuman strength”,
blue);
flash = new Superhero (“Barry Allen”, “fastest”, red);

superman.protectPeople(); //use object to perform a
function
flash.fightBadGuys();
```

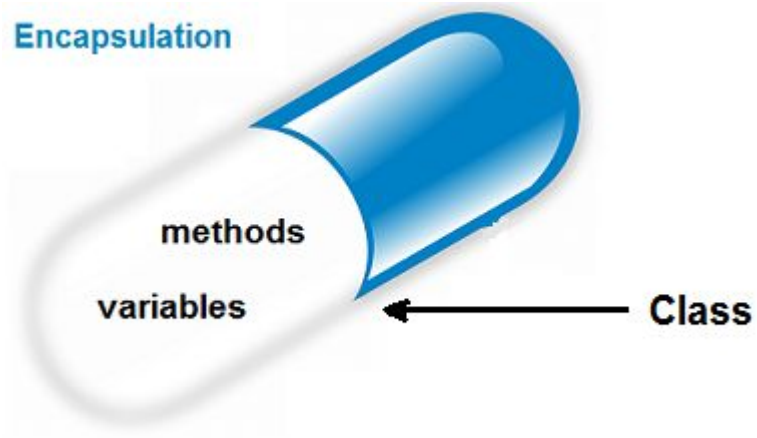
```
class Superhero {

        String realName; //variables
        String power;
        Color suit;

        //methods or functions
        void protectPeople() {-------}
        int fightBadGuys() {-------}
        void disguise() {-------}

        //constructor
        Superhero(String realNameX,
        String powerX, Color suitX){

                realName = realNameX;
                power = powerX;
                suit = suitX;
        }
}
```

```
Superhero superman; //declare the object
Superhero flash;

//initialize
superman = new Superhero (“Clark Kent”,
“superhuman strength”, blue);
flash = new Superhero (“Barry Allen”, “fastest”, red);

//use object to perform a
function
superman.protectPeople();
flash.fightBadGuys();
```

Principles of OOP

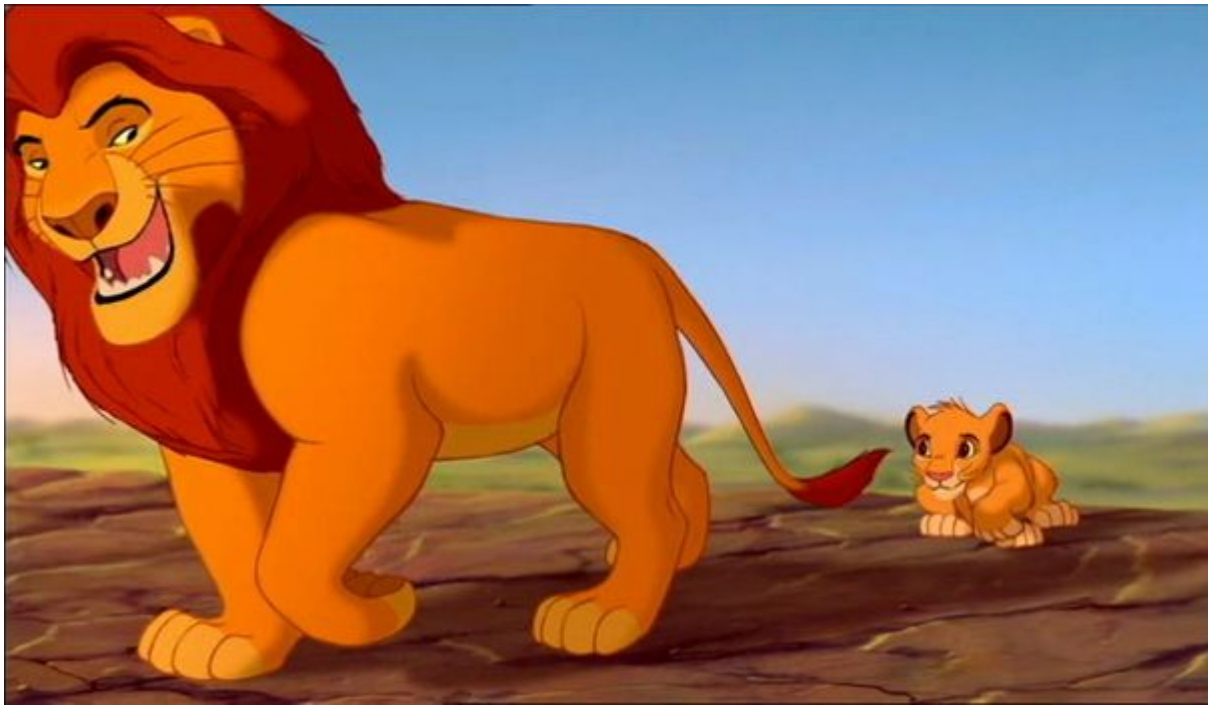
1. Encapsulation
2. Inheritance
3. Polymorphism

Encapsulation



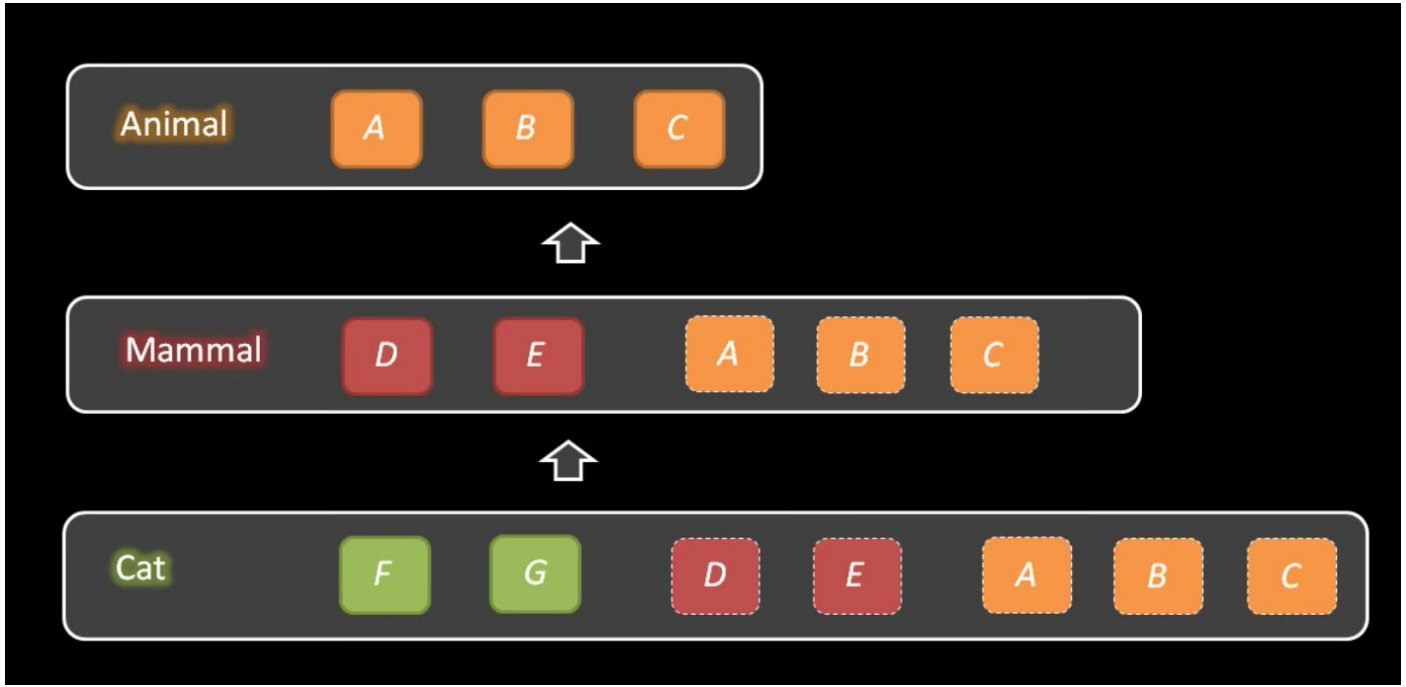
Inheritance

Base class



Derived class

Inheritance



Polymorphism

ArrayList

Stores a variable number of items

Unlike arrays, it's easier to add and remove items

Methods used with ArrayLists: size()
get()
add()
remove()

Let’s Code!

Homework

Work on Midterm!