Classes and objects

Class - defines a category which share common characteristics

Examples:

A cupcake class might store frosting flavor, cake flavor, and topping type

A song class might store the title, artist, album, and year written

Object - an instance of a class

Examples:

A strawberry cupcake is an instance of a type of cupcake

“Take on me” by A-Ha is an instance of a type of song

Classes

Classes are a technique for organizing complex programs

Classes allow us to define new types

- strings, integers, floats, booleans are **built-in types**
- Cupcake is an example of a new type (useful perhaps for a recipe program)
- Song is an example of a new type (useful perhaps for a music library program)

Classes

Classes group together data and functions

- data: what the class knows about (its state)
  - Example: Cupcake's data is frosting flavor, cake flavor, and topping type
- functions: what the class can do
  - Example: Cupcake might have a method for printing its recipe

Functions which belong to a class are called **methods**

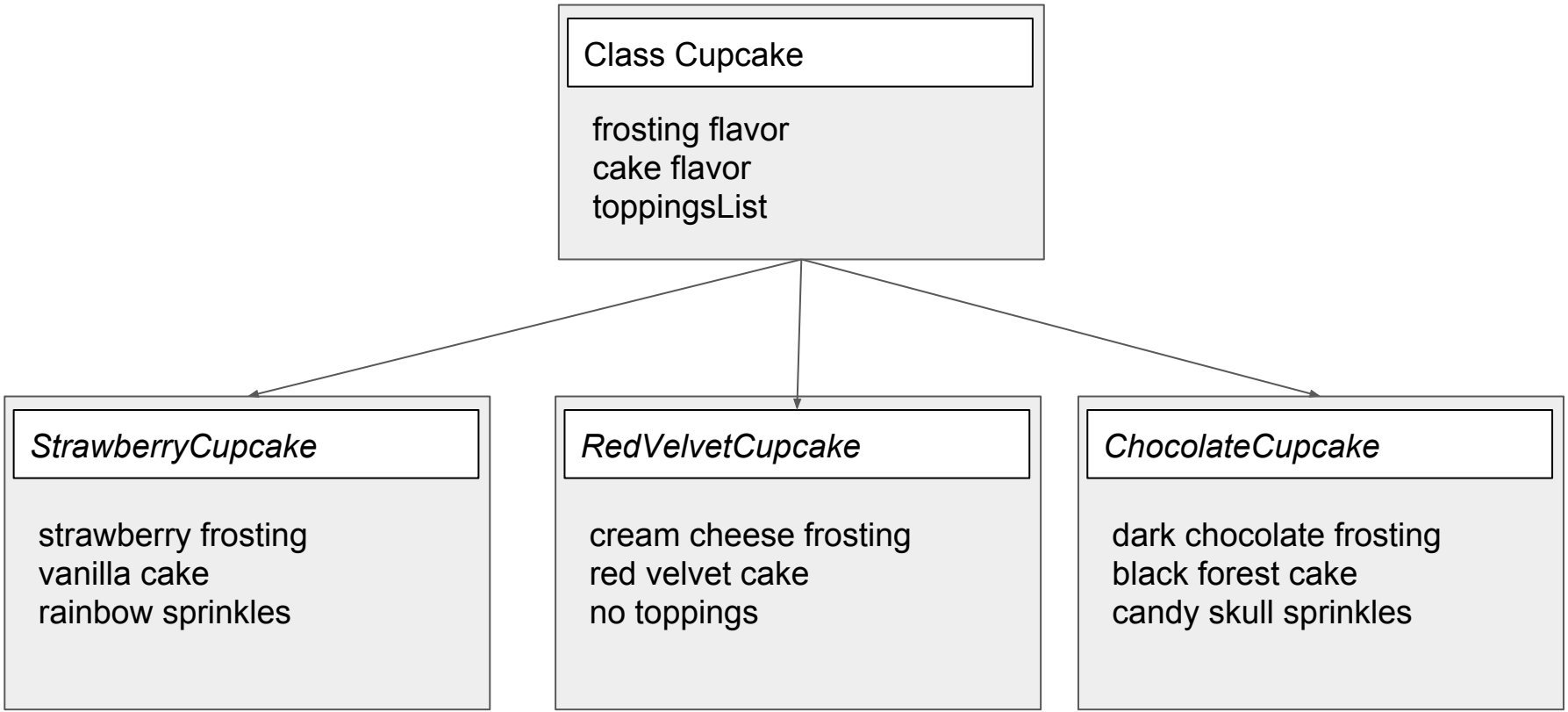
Classes

The process of designing a program in terms of classes is called **Object Oriented Programming (OOP)**

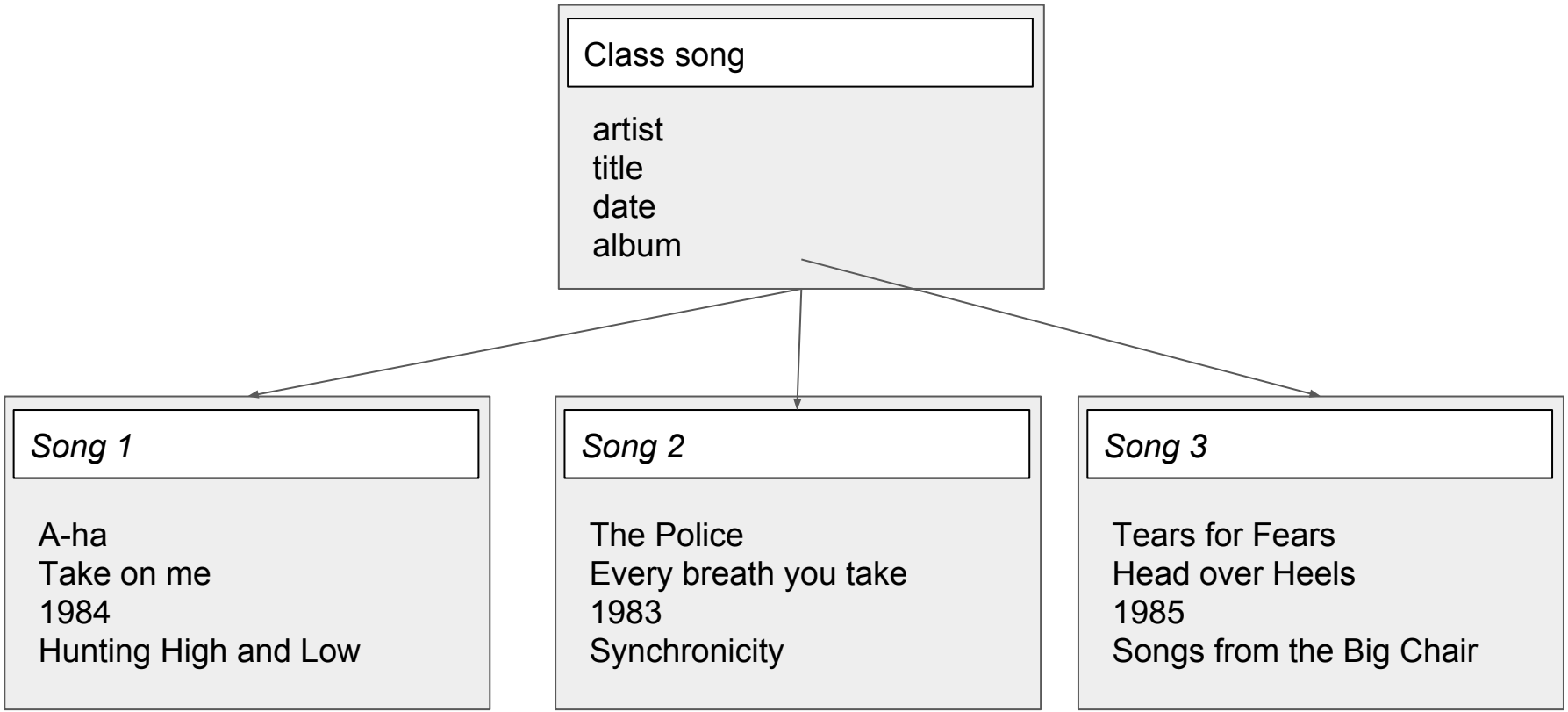
Like functions, classes provide

- abstraction - the user doesn’t need to know how the class works to use it
- modularity/ease of maintenance - classes are self-contained units which can be built and tested as components
- re-useability - write once and use forever

Class/object example: cupcake



Class/object example: songs for a music library



Using classes

Step 1: Create your object

Step 2: Manipulate your object via methods

```
>>> a = list()
>>> a
[]
>>> a.append("apple")
>>> a
['apple']
>>> a.append("banana")
>>> a
['apple', 'banana']
```

Creates an instance of the list class and stores it in the variable *a*

Calls the append() method to add the string “apple” to the end of the list

Class constructor

The function we use to create an instance of a class is called the **constructor function**

Syntax:

```
variable = ClassName(param1, param2, ….)
```

Examples:

```
a = list()

point = Point(50, 50)

circle = Circle(point, radius)
```

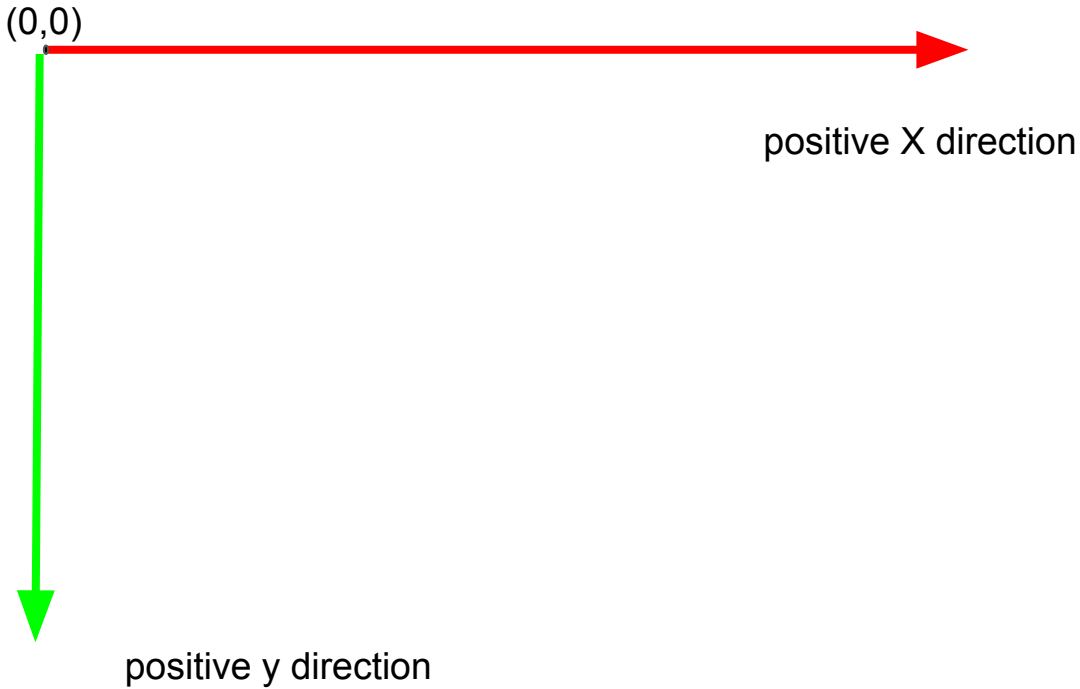
Graphics with Zelle

We will use Zelle to practice using classes

(Later in the semester we will learn how to define our own classes!)

Zelle defines classes for points, shapes (circles, rectangles, polygons), text, colors, a graphical window, and more…

Zelle coordinate system



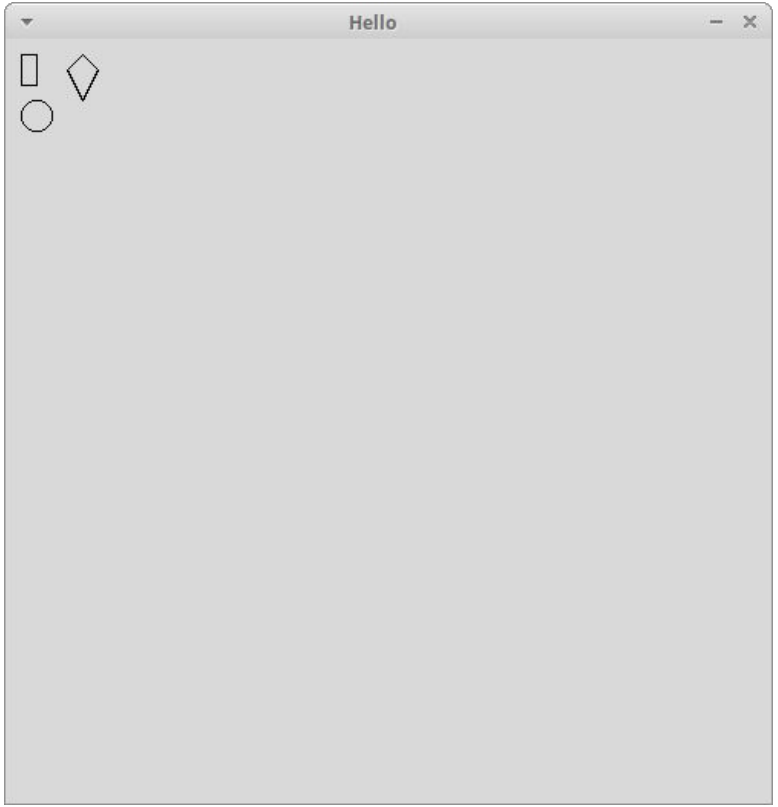
Exercise: simple scene

Create a window with width 500 and height 500

Create a circle at point (20, 50)

Create a polygon with points (50, 10), (60, 20), (50, 40), (40, 20)

Create a rectangle with top corner at (10,10) and bottom right corner (20,30)



Exercise: Simple scene