Object-Oriented Programming

Elements of Graphics
CS324e
Spring 2018
Objects in Code

❖ What are objects?
  ❖ A grouping of related functions and variables
❖ How does grouping these assist programmers?
  ❖ Provides code structure and organization
  ❖ Allows for more modular, higher level considerations
Classes

- Defines a group of related methods (functions) and fields (variables)
- Defines the behaviors and interactions of these methods and fields
- Outside classes do not need to consider implementation — just expected behavior
Object Instances

- Constructed based on the parent class’s specifications
- Multiple objects from the same class are independent
  - Can act (and be acted upon) in individual ways
- But objects still have same expected behavior even if they occupy different states
## Class Versus Object

<table>
<thead>
<tr>
<th>Car Class</th>
<th>Car Object</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name:</strong> Car</td>
<td><strong>a_car</strong></td>
</tr>
<tr>
<td><strong>Fields:</strong> make, model, color, speed</td>
<td><strong>make:</strong> Honda</td>
</tr>
<tr>
<td><strong>Methods:</strong> accelerate(), brake()</td>
<td><strong>model:</strong> Civic</td>
</tr>
<tr>
<td></td>
<td><strong>color:</strong> black</td>
</tr>
<tr>
<td></td>
<td><strong>speed:</strong> 0</td>
</tr>
</tbody>
</table>
class Spot {
    float x, y, radius;
    void display() {
        ellipse(x, y, radius, radius);
    }
}

Class Code Example
What’s Missing?

- Write a method that will “complete” our Spot class!
Constructors

- Block of code that is activated upon object instantiation
- Method always shares class name
- Can assign values to object fields
Multiple Constructors

Spot() {
    x = 50;
    y = 50;
    radius = 30;
}

Spot(float x, float y, float _r) {
    this.x = x;
    this.y = y;
    radius = _r;
}
Spot sp;
void setup() {
    size(100, 100);
    sp = new Spot();
    sp.x = sp.y = 50;
    sp.radius = 15;
}
void draw() {
    sp.display();
}

class Spot {
    float x, y, radius;
    Spot() {};
    void display() {
        ellipse(x, y, radius, radius);
    }
}
What does the keyword this mean?

```java
Spot(float x, float y, float r) {
    this.x = x;
    this.y = y;
    this.r = r;
}
```
Using Objects

- Each object from a class must be created using `new`:

```java
sp1 = new Spot();
sp2 = new Spot(75, 80, 15);
```

- Now we can display each object individually in `draw()`:

```java
void draw() {
    sp1.display();
    sp2.display();
}
```
Extending Class Functionality

- Fields represent meaningful object values
  - What might speed represent in Spot?
  - What might direction represent in Spot?
- Methods represent meaningful object behaviors
  - How could we use a move() method in Spot?
Class Files

- A single file can contain all of a program’s classes
- Multiple files can break up a program’s classes
- Multiple files provide modularity
- Easier for groups to work on code base
Using Multiple Files

1. Create main program \(\text{setup}()\) and \(\text{draw}()\) functions) in a sketch folder

2. Select “New Tab”

3. Give the file the name of the class it contains

4. Reuse class files by copying them to other sketch folders
Hands-on: Using Classes

❖ Today’s activities:

1. Implement the Spot class in a Processing sketch. Be sure that it is within its own file

2. Add a speed field and a move( ) method, so the spot's position can update

3. Create at least two Spot objects that start out with different positions and speeds