Classes

Object-Oriented Design

- Method for designing computer programs
- Consider "objects" interacting in the program
  - Example: a zoo, a gradebook

OOD Goals

- Robustness
  - Gracefully handle failures
- Adaptability
  - Evolve as necessary
- Reusability
  - Many programs use same piece of code

OOD Principles

- Abstraction
  - Abstract Data Types (ADTs)
  - Interfaces
- Encapsulation
  - Information Hiding
- Modularity
  - Easily plug together components

What is a class?

- Data and the methods that operate on that data – collectively called members
  - Example: bank account class
- Provide structure for organizing programs

Methods

- Typically, data (variables) declared private
- Methods operate on data
  - accessors – read data, provide access to data but do not change it
  - mutators – change data
    - examples from bank account, zoo???
  - constructor – builds a new object

Writing Classes

- Must be implemented in a file named *classname*.java
  - well…there are also inner classes

BankAccount Class

- `public BankAccount(double balance);`

- `public void withdraw(double amount);`
- `public void deposit(double amount);`
- `public double checkBalance();`

Creating and Using Objects

```
BankAccount b = new BankAccount(500);
//Type Name = new Type(constructor parameters);

//how would you withdraw funds?
```

Creating and Using Objects

```
//how would you withdraw funds?
b.withdraw(300);
object_name.method_name(param list);
```

Constructor

- Special-case function called when a new object is created
- Used to initialize member variables
  - Examples?
- Default constructor takes no parameters

Flight class

- Think about the design of a class to represent a flight…
  - Data members?
  - Methods?

Scope

- What is the scope of each of the variables you declared in your flight class?

static

- Static class member - a variable with scope the same as a class member
  - 1 per class, not per object
- Example - car serial number