CS171
Object Oriented Programming
Cognitive Perspective

- Humans understand things in terms of:
  - Attributes (sometimes called properties)
  - Behaviors

- Type of things share:
  - Common attributes with different values
  - Common behaviors customized to attribute values
Computing Perspective

- Classes: types of thing
- Objects: instances of classes
- Attributes: data common to classes with unique values per object
- Methods: functions that implement behaviors
Triggering Behaviors

- aka Invoking Methods
- Conceptually
  - Messages sent from one object to another
- Implementation
  - Calling methods
From an object-oriented purist perspective:

- Objects do not directly access attributes of other objects.
- They use accessor methods.
  * Inspectors: methods that look at attributes
  * Mutators: methods that change attributes
Creating Objects

- Special methods:
  - Constructor: create an object by instantiating a class
  - Destructor: destroy an object
Polymorphism

• Literally: many forms
• Variant methods (or operators) with the same name
• Distinguished by different numbers of types of arguments
C++ Objects

- Classes are like types similar to int, double, etc.
- Declarations instantiate classes into object
- Declarations invoke default constructors
- Alternate constructors (example of polymorphism) often exist
- Constructor methods named the same as the class
C++ Object Creation Examples

- `string name;` — creates empty string object
- `string name("Bob");` — creates string object containing “Bob”
- `string name = "Bob";` — another way to create and initialize
Invoking C++ Methods

- General form: `variable.method(arguments)`
- Examples:
  - `name.length()`
  - `name.find("ob")`
  - `name.substr(start, length)`
  - `fullname = firstname + " " + lastname;`