Labs posted. Due next Wed.

Processing: Scale of Variables.
  - Global vs. local scope
  - Objects

Define: The scope of a variable is the area of the program where it can be accessed.
  - Local scope: within a function
  - Global scope: within a file

Ex: Scope of Variables
Problem: Write a function that draws a stick figure at the point \((x, y)\).
Problem: draw a car

Ex. Car 1
All data types in Java fall into two categories:

- **Primitive types**
- **Reference types**

There are 8 primitive types in Java:

- **Name**
  - boolean
  - char
  - byte
  - short
  - int
  - long
  - float
  - double

- **Values**
  - true, false
  - 'a', 'b', 'c'

- byte: -128 to 127
- short: -2^15 to 2^15-1
- int: -2^31 to 2^31-1
- long: -2^63 to 2^63-1
- float: decimal #
- double: higher precision
We use the `class` keyword to create new data types. (These are reference types.)

For a car, a class is used to bundle

(1) the variables that constitute a car object; i.e. data:

```
xPos
yPos
speed
```

(2) the things that a car does:

```
displayCar()
moveCar()
```
(1) are **fields or member variables**.

(2) are **member methods or member functions**.

```cpp
class Car {

};
```

general form for class defn:

class Blah

// fields

// methods

class body

Blah is then a new (reference) data type.

Declaration:

Blah B
to create a Blah object use the `new` operator

```
B = new Blah();
```

allocation

```java
Car constructor
```

memory:

```
B
```

Blah

```
null val.
```

Blah object.

both declaration & allocation:

```
Blah B = new Blah();
```
Ex. Car 2: define car class

Ex. Car 3: define constructor