What you've learned: Computers (people, robots):

They work with input, stored, and output data.

Data is stored on named paper tickets, called variables.

What controls the work is called a computer program.
The sequential order of steps YOU write in your programs usually makes a big difference in what's computed. pen started thick pen was made thick here.
Java language related to people:

```java
int Yellow;
```

Get a ticket named Yellow on your desk, and promise only to use it for ints.

Geek speak: Declare Yellow to be the name of a variable of type int.
Java language related to people:

`Yellow = 33 ;`

Write 33 onto your Yellow ticket.

Geek speak: ASSIGN the variable (named) Yellow to have the value 33

The 33 (in the code) is called a literal value...it says literally what it is.
int Yellow;
int Pink;
Yellow = 33;
Pink = 37;
Pink = Pink - Yellow;

What's on the Yellow ticket?

(A) 33
(B) 37
(C) 4
(D) 5
Object Oriented Programming

After 1970 or so, people starting thinking:
It's sometimes easier to think of the “tickets” storing digital data as being real, active things like Turtle objects with potential behaviors called methods. (sometimes more confusing!)
Sad but true about Java:
Two kinds of tickets

**Object**

```
Turtle mt;
```

**Primative**

```
int Yellow;
```

On the ticket named Yellow is written a whole number.

On the Turtle is written:
- it's Color, whether pen is up or not,
- penWidth,
- x, y position,
- heading,
- which World it's in...
Certain kinds of software are very easy to write in an Object Oriented language.

```java
World w;
Turtle mt=new Turtle(w);
mt.forward(100);
mt.turn(90.0);
mt.forward(100);
mt.turn(90.0);
mt.forward(100);
mt.turn(90.0);
mt.forward(100);
```
World w;
Turtle mt=new Turtle(w);
mt.forward(100);    What's drawn?
mt.turn(90.0)      (A) circle
mt.forward(100);   (B) square
mt.turn(90.0);     (C) triangle
mt.forward(100);   (D) non-square
mt.turn(90.0);     rectangle
mt.forward(100);   (E) other
What's a class?
first thought: A bag to hold software.

After 2 lectures..
A container for an application program:

It specifies where the instructions begin. They begin at the top of the special

```java
public static void main(String a[])
{
    //start HERE!!
}
```
Next, you coded METHOD CALLS inside the main() method.

```java
public static void main(String a[])
{
    World w = new World();
    Turtle mt = new Turtle(w);
    mt.penUp();
    mt.moveTo(20,20);
    mt.penDown();
    mt.drawL(3.0);
    //the CALLER IS main
    //the CALLEE is drawL
}
```
the caller wrote on a calling card:
address to get to his Turtle

\textit{a copy of what's on ticket} \texttt{mt}

remember: code of main is: \texttt{mt.drawL(3.0)}

scale (size) number \texttt{3.0}

whom to tell the drawing is done

\textit{where in computer memory is the code of main}
What's a class?
  first thought: A bag to hold software.

After 4 lectures..
A container where YOU can code methods
for custom behavior you can program an
object to do.. Like drawL, drawX, etc.
are new behaviors for a Turtle.

```java
public void drawL(double scale)
{
  this.forward((int)*(100*scale));
}
```
Moving on to Pictures..

A 20-21\textsuperscript{st} century digital image is an array of (today) millions of little tickets where each one has 3 numbers scribbled on it.

Geeks call each ticket a Pixel (for Picture Element)
1,920 x 1,080 is the resolution of today's best HDTVs. This means the array is
1,920 Pixels wide by
1,080 Pixels high. How many tickets (each with 3 numbers)?

(A) a few thousands
(B) around 1,000,000
(C) around 2,000,000
(D) around 6,000,000
(E) billions and billions
How to succeed in CS w/o trying:
(1) Do what those 1940's computing women did: follow the computing instructions LINE-BY-LINE
    BEGINNING in the code block under
public static void main(String a[])

New for Lect09: When the computer runs a for-loop, the code right after the
for ( runs FIRST.
FACT #1 about for-loops: When the computer runs a for-loop, the code right after the for runs FIRST, & only once.

FACT#2: That can be ANYTHING that you might code anywhere else in a Java program!
When the computer runs a for-loop, the code right after the for runs FIRST, & only once.
Let's imagine, think about, plan, take ----

(all by herself)

out of her context (the for loop)!
int x;

Now alone, by herself.
class BadForDemonstrator
{
    public static void main(String a[])
    {
        for( int x; x < 10; x = x + 1 )
        {
            System.out.println(x);
        }
    }
}
for( int x ; 
    ..... ; ..... ) 
{ ..... }

Error:
BadForDemonstrator.java:5: variable x might not have been initialized
done first and only once in our for-loop

```c
int x = 0;
```

done first and only once in our for-loop
Whew, that was tedious...

.... to learn...

....but now....
UA 201 class of Spr12
KNOWS
for sure..
Dear Ms. Geek,

Please set up a for-loop having an int counter x that starts at 0.

--thanks.
That's easy:
for(int x=0;
    ???
    ??? ;
    ???
    )
{
    //you're welcome
class ForDemonstrator
{
    public static void main(String a[])
    {
        for(int x=0; x < 10; x = x + 1)
        {
            System.out.println(x);
        }
    }
}
```java
{ 
    System.out.println( x );
} 

Body of the for-loop
When *that body* gets to control the computer, the computer
(1) .. looks at ticket x
(2) .. sees what number is currently on it.
(3) (.. writes it on a calling card and calls method `println on obj. out`)
(4) .. prints that number
(5) (.. makes the printing spot go to the next line)
THAT'S ALL FOLKS!
System.out.println(x);

> java BodyRunner

> 0
class BodyRunner {
    public static void main(String a[])
    {
        int x = 0;
        {
            System.out.println( x );
        }
    }
}

> java BodyRunner
0
>
class PictureWorker {
    public static void main(String a[]) {
        String fileName = FileChooser.pickAFile();
        Picture myPicture = new Picture( fileName );
        System.out.println(myPicture);
        myPicture.show();
        myPicture.explore();
        //motorcycle's nose is at x=526, y=132
        for (int x = 526; x <= myPicture.getWidth()-1; x=x+1) {
            Pixel addrPix = myPicture.getPixel(x,132);
            System.out.println(addrPix);
            addrPix.setRed(255);
            addrPix = myPicture.getPixel(x,132+1);
            System.out.println(addrPix);
            addrPix.setRed(255);
        }
        myPicture.explore();
        myPicture.write("/tmp/Lect08Result.bmp");
    }
}
class PictureWorker
{
    public static void main(String a[])
    {
        String fileName = FileChooser.pickAFile();
        Picture myPicture = new Picture(fileName);
        System.out.println(myPicture);
        //myPicture.show();
        //myPicture.explore();
        for (int x = 526; x <=myPicture.getWidth()-1; x=x+1)
        {
            Pixel addrPix = myPicture.getPixel(x,132);
            System.out.println(addrPix);
            addrPix.setRed(0);
            addrPix.setGreen(0);
            addrPix.setBlue(0);
            addrPix = myPicture.getPixel(x,132+1);
            System.out.println(addrPix);
            addrPix.setRed(0);
            addrPix.setGreen(0);
            addrPix.setBlue(0);
        }
        myPicture.explore();
        //myPicture.write("/tmp/Lect08Result.bmp");
    }
}
Examples of for-loops
true or false?
for ( i ; false ; true )

true  KEEP GOING

false  finish

true or false

?
while(
    true or false?
)
while(true or false)

true KEEP GOING

false

finish

true or false?