CSI201
Lectures 05-06
Feb. 12-14 2013
Your RED ticket. Use this ticket:
To write a number on, to read a number to tell if it is 0, or to compare it to the number on another ticket, or to subtract it with another number from a ticket, and also, ERASE and replace with another number.

| red = 126; |
| green = 102; |
| blue = 165; |

Your GREEN ticket. Use this ticket:
To write a number on, to read a number to tell if it is 0, or to compare it to the number on another ticket, or to subtract it with another number from a ticket, and also, ERASE and replace with another number.

Your BLUE ticket. Use this ticket:
To write a number on, to read a number to tell if it is 0, or to compare it to the number on another ticket, or to subtract it with another number from a ticket, and also, ERASE and replace with another number.

Write your name here_________________________________________ Job title: Computer.
Mission: Execute, run, perform, do, demonstrate Euclid’s algorithm on parameter numbers 126, 102, 165
Your program, recipe, specific instructions, orders, directions, what to do EXACTLY:

1. Write parameter number 126 on your RED ticket.
2. Write parameter number 102 on your GREEN ticket.
3. Write parameter number 165 on your BLUE ticket.
Comment: The rest of the instructions are roughly "subtract like crazy" until you get two zeros.
But that is not clear enough for a computer.
4. Look over the numbers and pick any two that are NOT BOTH 0.
5. If step (4) failed because two tickets have 0’s on them, then STOP, and shout out the answer by reading the number on the remaining ticket. Also write it here_________
6. See which of the two non–zero numbers you picked is bigger than the other.
   If they are the same, just pick one of the two tickets with the common number.
7. Put your thumb on the ticket whose number you picked.
8. Subtract the OTHER picked number from the number your thumb is on. (The difference should be 0 if they are the same, and positive if they are different.)
9. Erase (or cross out) the number under your thumb and overwrite it with the difference, the number you got from the subtraction you did in step 8.
10. Now, go back to step (4). You will repeat steps (4) – (9) over and over until you get two zeros.

How instructions 1, 2, and 3 are expressed in Java. Java requires all variables to be “declared” but some other languages don’t.
public class undeclaredVarApp
{
    public static void main(String[] a)
    {
        System.out.println(red);
    }
}

A) Runs and prints 0
B) Runs and run-time error ("crash")
C) Won't compile: variable red might be uninitialized
E) Won't compile: can't find symbol variable red
public class undefinedVarApp
{
    public static void main(String[] a)
    {
        int red; //Sets up MEMORY!
        System.out.println( red );
    }
}

Every variable must be DECLARED before it can be used. "Declare" doesn't express well what it really means! It is a word of jargon. What is means is most important..

The declaration (like `int red;`) means

Two

1. BORROW SOME

   MEMORY think: Whiteboard space

   actions!

2. Make red symbolize that space.
public class undefinedVarApp
{
    public static void main(String[] a)
    {
        int red; //Sets up MEMORY!
        System.out.println(red);
    }
}

A) Runs and prints 0
B) Runs and run-time error ("crash")
C) Won't compile: variable red might be uninitialized
E) Won't compile: can't find symbol variable red
public class initializedVarApp
{
    public static void main(String[] a)
    {
        int red; // It sets up MEMORY!
        // A variable is like a ticket.
        // And, make red be its name.

        red = 126; // It means copy 126
        // onto the red ticket, erasing
        // any previous value.
        System.out.println( red );
    }
}
public class initializedVarApp
{
    public static void main(String[] a)
    {
        int red; //It sets up MEMORY!
        red = 126; //It means copy 126
            //onto the red ticket, ...
        System.out.println(red);
    }
}

A) Runs and prints 0
B) Runs and prints 126.
C) Something else happens.
public class repeatedVarWriteApp
{
    public static void main(String[] a)
    {
        int red; //It sets up MEMORY!
        red = 126; //It means copy 126
        red = 9;
        System.out.println( red );
    }
}

A) Runs and prints 0
B) Runs and prints 126.
C) Runs and prints 9
D) Runs and prints 117 (is 126-9)
E) Something else happens.
public class twoVarApp1{
public static void main(String[] a)
{
    int red;
    int green;
    red = 126;
    green = 102;
    System.out.print(red);
    System.out.println(" "+green);
}

A) Runs and prints 228 (is 126+102)
B) Runs and prints 126 102
C) Runs and prints 102 126
D) Runs and prints 126 126
E) Runs and prints 102 102

Two spaces really.
red = green;  // It means COPY.

The copying is ALWAYS
from the RIGHT
to the LEFT

RIGHT  hand  side of the =
LEFT   hand  side of the =
Look over the Plan Edition 01

MyProgrammingLabbers--stay tuned if it put you in the wrong section
plan: reflect same asg. into Spr12 ask HELP!! to Pearson..
Read Plan Edition 01

MyProgrammingLabbers-- if it put you in the wrong section 
...
I reflected the SAME asg. into Spr12
  Just do
    Albany 201 MyProgHW1
    Albany 201 MyProgHW2

won't bother Pearson yet..
A variable IS a named storage location (that means a physical spot in your computer's memory (RAM) hardware) in the computer's memory. The data IN it ("it" is the variable) (the data is the variable's current value) MAY CHANGE.
public class twoVarApp1{
public static void main(String[] a)
{
int red;
int green;
red = 126;
green = 102;
System.out.print( red );
System.out.println(" "+green);
}

A) Runs and prints 228 (is 126+102)
B) Runs and prints 126 102
C) Runs and prints 102 126
D) Runs and prints 126 126
E) Runs and prints 102 102

Two spaces really.
The essence (and challenge) of programming.  

You (programmer, PEOPLE) **write** instructions  
(like **WRITING** A COOKBOOK—**NOT COOKING** yourself!)  
that tell a computer  
(robot or person)  
what to do with VARIABLES.  

**options:**  
Make and name them (“declare”).  
Store literal values in them.  
Use their current value for something like how long a line to make a Turtle draw.  
Calculate and **CHANGE** their current values.
red = green; // It means COPY.

The copying is ALWAYS from the RIGHT hand side of the = to the LEFT hand side of the =

from the RIGHT to the LEFT
```java
public class twoVarApp3{
    public static void main(String[] a) {
        int red;
        int green;
        red = 102;
        green = 126;
        red = green; // = = means COPY!!
        System.out.print( red );
        System.out.println( " " + green );
    }
}
```

B) Runs and prints 126  102
C) Runs and prints 102  126
D) Runs and prints 126  126
E) Runs and prints 102  102
red = green; // It means copy.

The copying is ALWAYS from the LEFT hand side of the = to the RIGHT hand side of the =.
public class twoVarApp3{
public static void main(String[] a){
    int red;
    int green;
    red = 102;
    green = 126;
    red = green; // = means COPY!!
    System.out.print(red);
    System.out.println(" "+green);
}
B) Runs and prints 126 102
C) Runs and prints 102 126
D) Runs and prints 126 126
E) Runs and prints 102 102
public class twoVarApp4 {
    public static void main(String[] a) {
        int red;
        int green;
        red = 102;
        green = 126;
        green = red; // = means COPY!!
        System.out.print( red );
        System.out.println(" "+green);
    }
}

B) Runs and prints 126  102
C) Runs and prints 102  126
D) Runs and prints 126  126
E) Runs and prints 102  102
Real programmers figure it on PAPER or Dry-erase boards (show the class how!)
```java
public class twoVarApp5{
public static void main(String[] a)
{
    int red;
    int green;
    red = 102;
    green = 126;
    green = green - red;
    System.out.print(  red  );
    System.out.println(" "+green);
}
A) Prints 102 126
B) Prints 102 102
C) Prints 102 24
D) Prints 102 -24
```
Something NEW (and red green and blue)

Next week's lab (starts Monday!) you will retrieve your picture and make a PURPLE SPOT on your nose

Reading (BEFORE LAB)
UA Custom 4.1 and 4.2
G&E 4.1 and 4.2 (coincidence!)
Digital Pictures...What they are...
FOR PROGRAMMERS!! paper array
 cut and assembled..“array”

9 x 9 pixel resolution
digital image!
Diagram to clarify G&E's Fig. 4.4

\[ \begin{array}{cccc}
0 & 1 & 2 & 3 \\
15 & 12 & 13 & 10 \\
9 & 7 & 43 & 23 \\
6 & 13 & 15 & 16 \\
\end{array} \]

\( x, \text{horizontal coordinates} \)
\( y, \text{vertical coordinates} \)

\((2, 1)\) locates the variable whose value is 43.
Picture (gray scale) actually stored in computer memory: 1-dimensional array.
Here are \( y \) full rows. The location is \((x, y)\).

Number (how many) of pixels in each row is

\[
\begin{align*}
&A(640) \\
&B(480)
\end{align*}
\]
Here are \( y \) full rows.

Location is \((x, y)\)

Number (how many) of pixels in each row is

(A) 640

(B) 480
How to figure out translating a (x, y) (2-dimensional) LOCATION (coordinates of one Pixel) into one integer (1-dimensional array subscript) that LOCATES the same Pixel in a 1-dimensional array of Pixels.

Pixel[] pxArray = pRef.getPixels();

A method name spelled with an s.
How many Pixels are in the red rectangle?

640 \times \text{Y} \leftarrow \text{Multiply in Java}
How many Pixels are before $(x, y)$?

$640 \times y + x$
public class twoVarApp6{
public static void main(String[] a) {
    int red;
    int green;
    red = 102;
    green = 126;
    green = red - green;
    System.out.print(red);
    System.out.println(" " + green);
} } A) Prints 102 126
B) Prints 102 102
C) Prints 102 24
D) Prints 102 -24
Doing a Project Like Project 01
Let it be letterArtMakingApp

Let's make it make a Z below a T

Planning
Quality HOW IT COUNTS in your Proj01 grade.
Enhancing a Turtle to a LiteraryTurtle (LIKE LAB03)
KIND OF DETAIL for revising, writing and reading computer programs.
More on Lab03 if there is time.
CSI201
Lectures 05-06
Feb. 12-14 2013
Demonstrated on paper.

A Variable means a named storage location, which really is like a piece of paper or a “ticket” where you can write data, copy the data from, read and use it in a calculation, and erase/overwrite the current data with new data

We WILL get to “declaring” soon, but the existence and read and changeable nature of variable is much more critical to understanding than whether or not a particular language (Java or other) requires all variables to be declared or not.

What’s another problem? RED is spelled with CAPITALS in the box but lowercase red in the Java!

In Java and most other modern languages, case makes a difference!

When all three tickets have non-zero numbers, the directions are ambiguous. The person or robot doing the computing has 3 choices for which two tickets to use for the subtraction and rewrite steps! It's taught to Ph.D. students in computer science (but not to beginners, usually) that some good algorithms or programs allow arbitrary choices sometimes! The choices can make the computer do different particular calculations when different choices are made. The algorithm is still good because the same the same ANSWER comes out no matter which choices are made.

The algorithm above, with the arbitrary choices, is one of the first devised by humans. It is named the subtractive version of Euclid's algorithm. Its answer is always the greatest common divisor of the input numbers. The same Euclid wrote out the books of geometry that base the subject of geometry you studied in high school.
The text with yellow background was literally copied from the two particular error messages that DrJava prints in different situations. Textual differences like this are the kind of detail programmers attend to. These two messages do mean different things. You are here to learn what these things are!

Tips for dealing with syntax errors: Worry ONLY about the very FIRST one; don't try to fix the others. Carefully read everything the error message says! Also carefully read the line (also highlighted in yellow) of code where the error is detected.

Re-compile IMMEDIATELY after you think you figured it out and tried to correct the error.
public class undefinedVarApp
{
    public static void main(String[] a)
    {
        int red; //Sets up MEMORY!
        System.out.println(red);
    }
}

Every variable must be DECLARED before it can be used. “Declare” doesn’t express well what it really means! It is a word of jargon. What is means is most important.

The declaration (like int red;) means Two different actions!

1. BORROW SOME MEMORY think: Whiteboard space
2. Make red symbolize that space.
```java
public class undefinedVarApp {
    public static void main(String[] a) {
        int red; //Sets up MEMORY!
        System.out.println(red);
    }
}
```

A) Runs and prints 0
B) Runs and run-time error ("crash")
C) Won't compile: variable red might be uninitialized
E) Won't compile: can't find symbol variable red
public class initializedVarApp
{
  public static void main(String[] a)
  {
    int red; //It sets up MEMORY!
    //A variable is like a ticket. 
    //And, make red be its name.

    red = 126; //It means copy 126 
    //onto the red ticket, erasing 
    //any previous value.
    System.out.println( red );
  }
}

public class initializedVarApp
{
    public static void main(String[] a)
    {
        int red; // It sets up MEMORY!
        red = 126; // It means copy 126
        // onto the red ticket, ...
        System.out.println(red);
    }
}

A) Runs and prints 0
B) Runs and prints 126.
C) Something else happens.
public class repeatedVarWriteApp
{
    public static void main(String[] a)
    {
        int red; //It sets up MEMORY!
        red = 126; //It means copy 126
        red = 9;
        System.out.println(red);
    }
}

A) Runs and prints 0
B) Runs and prints 126.
C) Runs and prints 9
D) Runs and prints 117 (is 126-9)
E) Something else happens.
public class twoVarApp1{
public static void main(String[] a)
{
    int red;
    int green;
    red = 126;
    green = 102;
    System.out.print(red);
    System.out.println(" "+green);
}
}

A) Runs and prints 228 (is 126+102)
B) Runs and prints 126 102
C) Runs and prints 102 126
D) Runs and prints 126 126
E) Runs and prints 102 102

Two spaces really.

READ in the textbooks about printing Strings and using the + symbol to concatenate two strings. Yes, + is the same + symbol that means addition when applied to number variables or literals. When + is after a String, like the literal string " " (the String of 2 spaces), a number after it is converted to decimal characters and stuck on after the two spaces.
red = green; // It means COPY.

The copying is ALWAYS from the RIGHT to the LEFT.
Look over the Plan
Edition 01

MyProgrammingLabbers--
stay tuned if it put you in the wrong section
plan: reflect same asg. into Spr12
ask HELP!! to Pearson..
Lecture 06
Feb 14, 2013
Read Plan Edition 01

MyProgrammingLabbers-- if it put you in the wrong section

....

I reflected the SAME asg. into Spr12
Just do
Albany 201 MyProgHW1
Albany 201 MyProgHW2

won't bother Pearson yet..
A variable IS a named storage location
(that means a physical spot in your computer's memory (RAM) hardware)
in the computer's memory.
The data IN it
(“it” is the variable)
(the data is the variable's current value)
MAY CHANGE.
public class twoVarApp1{
public static void main(String[] a)
{
    int red;
    int green;
    red = 126;
    green = 102;
    System.out.print(red);
    System.out.println(" "+green);
    }

A) Runs and prints 228 (is 126+102)
B) Runs and prints 126 102
C) Runs and prints 102 126
D) Runs and prints 126 126
E) Runs and prints 102 102

Two spaces really.

READ in the textbooks about printing Strings and using the + symbol to concatenate two strings. Yes, + is the same + symbol that means addition when applied to number variables or literals. When + is after a string, like the literal string " " (the String of 2 spaces), a number after it is converted to decimal characters and stuck on after the two spaces.
The essence (and challenge) of programming..

You (programmer, PEOPLE) write instructions
(like WRITING A COOKBOOK—NOT COOKING yourself!)
that tell a computer (robot or person) what to do with VARIABLES.
options:
Make and name them (“declare”).
Store literal values in them.
Use their current value for something like how long a line to make a Turtle draw.
Calculate and CHANGE their current values.
red = green; // It means COPY.

The copying is ALWAYS from the RIGHT hand side of the = to the LEFT hand side of the =
public class twoVarApp3{
    public static void main(String[] a)
    {
        int red;
        int green;
        red = 102;
        green = 126;
        red = green; // = means COPY!!
        System.out.print(red);
        System.out.println(" "+green);
    }
}

B) Runs and prints 126 102
C) Runs and prints 102 126
D) Runs and prints 126 126
E) Runs and prints 102 102
red = green; //It means COPY.

The copying is ALWAYS from the RIGHT to the LEFT
public class twoVarApp3{
    public static void main(String[] a)
    {
        int red;
        int green;
        red = 102;
        green = 126;
        red = green; // = means COPY!!
        System.out.print( red );
        System.out.println( " " + green);
    }
}

B) Runs and prints 126 102
C) Runs and prints 102 126
D) Runs and prints 126 126
E) Runs and prints 102 102
public class twoVarApp4{
public static void main(String[] a)
{
    int red;
    int green;
    red = 102;
    green = 126;
    green = red; // = means COPY!!
    System.out.print( red );
    System.out.println(" "+green);
}
}
Real programmers figure it on PAPER or Dry-erase boards (show the class how!)
public class twoVarApp5{
public static void main(String[] a)
{
    int red;
    int green;
    red = 102;
    green = 126;
    green = green - red;
    System.out.print(red);
    System.out.println(" "+green);
}
}

A) Prints 102  126
B) Prints 102  102
C) Prints 102  24
D) Prints 102  -24
Something NEW (and red green and blue)

Next week's lab (starts Monday!) you will retrieve your picture and make a PURPLE SPOT on your nose.

Reading (BEFORE LAB)
UA Custom 4.1 and 4.2
G&E 4.1 and 4.2 (coincidence!)
Digital Pictures...What they are... FOR PROGRAMMERS!! paper array cut and assembled..“array”

9 x 9 pixel resolution digital image!
Diagram to clarify G&E's Fig. 4.4

\[ \begin{array}{cccc}
0 & 1 & 2 & 3 \\
15 & 12 & 13 & 10 \\
9 & 7 & 43 & 23 \\
6 & 13 & 15 & 16 \\
\end{array} \]

\( \chi, \text{horizontal coordinates} \)

(2, 1) locates the variable whose value is 43.
Picture (gray scale) actually stored in computer memory: 1-dimensional array.
Here are \( y \) full rows.

Location is \( (x, y) \)

Number (how many) of pixels in each row is

A) 640
B) 480
Here are your full rows. Location is $(x, y)$.

Number (how many) of pixels in each row is

(A) 640
(B) 480
How to figure out translating a (x, y) (2-dimensional) LOCATION (coordinates of one Pixel) into one integer (1-dimensional array subscript) that LOCATES the same Pixel in a 1-dimensional array of Pixels.

```java
Pixel[] pxArray = pRef.getPixels();
```

A method name spelled with an s®
How many Pixels are in the rectangle?

\[ 640 \times y \]
How many pixels are before $(x, y)$?

\[640 \times y + x\]
public class twoVarApp6{
public static void main(String[] a)
{
    int red;
    int green;
    red = 102;
    green = 126;
    green = red - green;
    System.out.print( red );
    System.out.println(" "+green);
}
}  
A) Prints 102 126
B) Prints 102 102
C) Prints 102 24
D) Prints 102 -24
Doing a Project Like Project 01
Let it be letterArtMakingApp

Let's make it make a Z below a T

Planning
Quality HOW IT COUNTS in your Proj01 grade.
Enhancing a Turtle to a LiteraryTurtle (LIKE LAB03)
KIND OF DETAIL for revising, writing and reading computer programs.
More on Lab03 if there is time.