Final Review Topics Fall 2012 ICSI 201
Chromakey—Project

- Textbook section: HOW DOES IT WORK??
- Loop to get at Pixels and their values in one Picture, get at their color numbers
  - And get at Pixels in another Picture
- If statement to decide whether to change the color of a Pixel
  - Lecture and Textbook material on if, and if/else statements.
- GE methods to get at Pixels, get a Pixel's color, and change a Pixels color.
Code to rasterize curves..Lab03
Work with one Pixel at a time Lab04

• Idea leads singly nested loop to rasterize a line (actually put a line in a Picture, Pixel by Pixel) or a rectangle, with a doubly nested loop.

• Such loops can be in parametrized methods, like the blackenRect method live coded.

• Coordinates can be involved in mathematical calculations, like to plot the trajectory of a thrown ball or other projectile.
Distinctions expressed by nesting

• One computation:

```plaintext
for( .. ; .. ; .. )
{
    DO SOMETHING before the inner for loop
    for( ... ; ... ; ... )
    {
        DO SOMETHING like blacken with ONE Pixel
    }
}
```
Distinctions expressed by nesting

- A different computation:

  ```
  for( .. ; .. ; .. )
  {
    DO SOMETHING like blacken with ONE Pixel
  }
  for( ... ; ... ; ... )
  {
    DO SOMETHING like blacken with ONE Pixel
  }
  ( I just moved one little } UP 5 lines!)```
Boolean and other values

- Boolean value means the set consisting of true and false (only two values in the set!)
- Only expressions with Boolean value are allowed in the ( )'s of if and while statements.
- Other primitive Java types: int and double
  - What they are good for: For example, ints but not doubles for array index values
  - Examples of the many elements in their sets:
    - Typical numbers without decimal points
    - Typical numbers with decimal points.
Boolean values

• Expressions for computing Boolean values
  – Example:
    
    boolean X;
    X = ( sc.nextInt() > 100 );

• Boolean variable to control a loop
  – Example:
    
    boolean keepShopping = true;
    while( keepShopping )
    {
      /* Get input */
      if( <a thing to buy> )
      {
        ... } else { keepShopping = false; }
    }
Loops, calculations, if statements

- Making a method to draw a rectangle on a Picture, or change a rectangle-full of Pixels.
- Parameters
- Loops
- Calculations (What should the new color of Pixel be? it's color be?)
- Blurring: compute the color intensities as weighted averages.
- If statements
Chapters 4, 5, 6

- The heart of the subject in the context of digital images (Pictures)
- Locating data in
  - Pictures by x,y location (2 indices) of Pixels
  - Arrays by one index.
- Calculating from and/or changing/storing data in an Picture or an array
- Loops
- Calculations
- Conditionals
- COMBINATIONS of Loops, calculations, conditionals
Intro to Image Processing

• Blur and (edge detection, etc.) are separate examples from Chapter 6 (modifing Pixels)

• Advanced topic(not done): Edge detection ALONE:
  – is TWO STEPS for each pixel
    • Compute absolute difference between brightness of Pixel and the Pixel above it
    • Detect line if the difference is big, don't detect if small
  – To research this, it's good to SEE the result of EACH STEP SEPARATELY

• COMPARE:
  – Edge detection in original picture
  – Edge detection in blurred original
Intro to Image Processing: Proj 03

- Blur and edge detection are separate examples from Chapter 6 (modifing Pixels)

- HOW DOES BLURRING WORK?
  - Explain in English and/or example of the calculation, NOT by regurgitating code.
static vs instance methods

Java static (also called class) methods are a good choice when the method does not have an obvious OBJECT to be CALLED ON.

Static methods cannot use “this” within their bodies...that makes no sense because “this” refers to the OBJECT the method is CALLED ON.
Methods returning values (or not)

Whether a method be static or instance, you might design it to either return a value or NOT return a value.

Topic: HOW to code (1) using the return value (easy: USAGE ( methodCall(...) )
(2) what value to return (easy: put it in the return statement in the method body)
My AlbumTester.java code

- User interface loops
  - Prompting for input
  - Reading input into a variable
  - if's to decide what to do
Last 2 Labs: Temperature and Stock Price record plotting and analysis

- Loops to print, process, search information stored in an array.
House, Student, Album classes

- Introduction to making your own classes: Textbook chapter 11
- The Mad Ph.D. Video:
  - A class as a BLUEPRINT for making objects, NOT an object itself!
  - What the new operation does.
  - What the new operation returns, and how to use what it returns
- Making your own class continues with Album of Project 05.
iClicker question
What does the Java new operation return?

(A) News that a new object was built.
(B) An actual new object.
(C) An address, location or reference to a new object.
(D) An address, location or reference to an old object that was built before the new operation was started.

See "Factory Pattern" for an alternative to new
iClicker question
What can your program the computer to do with the address, location or reference that a new operation just returned after it made an object?

(A) Print the address.
(B) Call a method ON the object.
(C) Save or copy the address into a reference ticket or variable.
(D) Call a method with the address as a parameter.
(E) B, C, and D but NOT (A)
Project 05

- Practice making and using **fields**, which are **variables inside objects**.

- **Constructor methods**
  - Study Ch. 11
  - relationship with new
  - What they are good for: Force field values to start out right.

- **Methods to add and display Pictures**
  - Different polymorphic forms of addPicture (See Ch. 11)
Project 05

- Accumulating data from step after step after step:
  - Track where to start copying the next Picture \( \texttt{nextX} \), after copying the previous.
  - Count of Pictures given and where to save next one in an array \( \texttt{nPictures} \)
  - Sum of the widths
  - Maximum of the heights
Project 04

• Coding to an interface SOMEBODY ELSE (the Prof!) gives you
  - Class MUST be named Album
  - It MUST have Album(int,int) and Album(int) constructors. (Each one earns you 50%)
  - It MUST have addPicture(Picture) and getResult() methods, that work as specified.
  - An example (AlbumTester.java) of user conversation code that RELIES on your coding the Album class properly.

• Classes you write that way are unit testable
Unit Testing

● Classes you write that way are **unit testable**

● That means they can be tested with unit tests.

● Java and other software development support systems have “frameworks” or infrastructure for defining and then running unit tests automatically
  
Revision Control, Version Histories

- It's modern practice for all serious software developers and large web site writers.
- Especially important when a possibly global team collaborates.
- Web links:
  - GIT, Mercurial (distributed)
  - CVS, Subversion (client-server)
- You can confidently DELETE instead of commenting out code you'll probably not use.
Themes..

• Main 201 goal: When given a sufficiently detailed description, or her/his own conception, of what a computer can do and a strategy for doing it, the student shall write a program that makes the computer do it by implementing the given strategy.

• Dependencies: What data must be input or computed FIRST, before other data it depends on can be computed or output.
  - Help you figure out a program that makes the computer compute in a correct ORDER.
LOCATION!!!

• Learning goals for data structures course readiness:
  – variables, arrays, control statements, methods, classes, basic problem solving
  – Solve program reading and writing problems in which some int variables are used for ARRAY INDEXING (and locating Pixels by coordinates). Examples: Finding index of largest and smallest elements, sorting the array, finding the candidate's NUMBER given the candidate's NAME, etc.

  Goal: You become ready to deal with pointer or reference and/or array index DATA whose purpose in to locate OTHER DATA.
Java arrays

• A Java array is a Java object, (like a Picture, Turtle, Pixel, Album, Stock, Temps, etc.)

• SO... it MUST be made (instantiated) by `new`
  - (code like `= { 2, 4, 6, 8 }` is just shorthand.)

• AND a reference variable must be used to store its address (otherwise, it's useless).

• The length of the array must be “known” (stored in the computer) FIRST, before the actual array can be made; and the array length CANNOT BE CHANGED.
How to make and use a Java array

- Declare a reference variable ready for the computer to write the address:
  - `int refToArray[];`
  - `//NEVER PUT THE LENGTH IN!`

- Having ALWAYS decided or computed the length, actually MAKE (instantiate) the array:
  - `refToArray = new int[366];`

- Having already decided or computed an index value:
  - Store data in an element: `refToArray[3] = 7;`
  - Retrieve data from an element:
    - `System.out.println(refToArray[3]);`
Similarity to making & using objects

- Declare reference variables:
  - `Stock myStock; Stock yourStock;`

- ACTUALLY MAKE THE OBJECTS:
  - `myStock = new Stock(); yourStock=new Stock();`

- Use objects by calling methods:
  - `myBill.addPrice(603.5);`
  - `yourBill.addPrice(34.3);`
Java arrays

• Given the array and index value, you can get at the element FAST; NO looping, searching, etc.

• The length of the array must be “known” (stored in the computer) FIRST, before the actual array can be made; and the array length CANNOT BE CHANGED.

• Why is this important?
  – Other languages, add-ons (ArrayList), and data structure techniques remove this limitation.