LAST CLASS! You made it! (Almost… 😊)

You should have gotten an email with your final exam seating assignment (let us know if not).

All exams are in the same classroom EXCEPT 5pm section is in the Rec Gym
About the Final

• Like a larger in-term exam

• Same format but 3-4x as long

• You will have 3 hours, but we will aim to create an exam that the average student can finish in 2 hours

• We will give you method headers for common methods (but not descriptions of what they do)

• What will be on it? You tell me…
Major topics in CSE 8A

• With your group, list as many topics as you can think of for CSE8a.
Studying for the final

1. Redo (i.e., write out, on paper):
   - A. PI (clicker) questions
   - B. Lab quiz questions
   - C. Reading quiz questions
   - Do this WITHOUT looking at the book. If you get stuck, STOP. Reread the relevant material in the book, maybe peek at the answer, then put the book away and try again

2. Get together with your friends (or post on Piazza for a group) and come up with new study questions. Then do those (using the same method)
How many objects are created in this code?

World world1 = new World(200,100);
Turtle maria = new Turtle(25, 25, world1);
Turtle jose = new Turtle(100, 50, world1);
maria.forward(25);
jose.forward(10);
maria.turnLeft();
maria.forward(50);
jose.forward(5);

A. 1
B. 2
C. 3
D. 4
Re-naming…!

```c
>> int x = 42;
>> int y = x;
>> x = 101;
>> S.o.p(x);
>> S.o.p(y);
```

What values are displayed for x and y?

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. 42</td>
<td>42</td>
</tr>
<tr>
<td>B. 101</td>
<td>42</td>
</tr>
<tr>
<td>C. 101</td>
<td>101</td>
</tr>
<tr>
<td>D. None of these</td>
<td></td>
</tr>
</tbody>
</table>

When in doubt, draw it out!!
What is the output of this code?

```java
int x = 3;
Turtle y = 2;
System.out.println(x == 3);
int z = x + y;
System.out.println(z);
```

A) 3   B) true
5       5

C) 3   D) true
false    false

E) None of the above.
What does this code draw?

World world1 = new World(200, 100);
Turtle maria = new Turtle(25, 25, world1);
Turtle jose = new Turtle(100, 50, world1);
maria = jose;
maria.forward(50);
jose.turn(90);
jose.forward(50);
Why is this code incorrect?

Assume this code exists inside the Turtle class in the Turtle.java file

```java
public void drawSquare()
{
    turtle1.turnLeft();
turtle1.forward(100);
turtle1.turnLeft();
turtle1.forward(100);
turtle1.turnLeft();
turtle1.forward(100);
turtle1.turnLeft();
turtle1.forward(100);
}
```

A. Nothing is incorrect

B. Return type is wrong

C. There needs to be a parameter

D. turnLeft should be turnRight

E. use of turtle1 is incorrect
Primitives vs. Objects: Review

- What does the following code print?

```java
int remzisAge = 19;
int rominasAge = 25;
remzisAge = rominasAge;
rominasAge = 20;
print( "Remzi is " + remzisAge + " and Romina is " + rominasAge )
```

A. Remzi is 19 and Romina is 25
B. Remzi is 20 and Romina is 20
C. Remzi is 25 and Romina is 20
D. Remzi is 25 and Romina is 25
E. Remzi is 19 and Romina is 20
Random Numbers

HINT: Random is a class that works as a generator of random numbers.

• What might the following code print?

```
import java.util.Random;

// declare a generator for random numbers
Random generator = new Random();

// generate a random number
int value1 = generator.nextInt(10);
double value2 = generator.nextDouble();

System.out.println(value1);
System.out.println(value2);
```

We need to IMPORT the package "explaining" what the class Random is!

1) Solo: (20 sec)
2) Discuss: (1 min)
3) Group: (20sec)

• What might the following code print?

A. 10 and 1.0
B. 1 and 2.0
C. 8 and 0.5
D. 8.5 and 0.5
E. More than one of these are possible
What does this code do?

Pixel[] pixelArray = this.getPixels();
int value = 0;
for (Pixel pixelObj: pixelArray)
{
    value = pixelObj.getGreen();
    pixelObj.setBlue(value);
}

A. Decreases the blue component of a picture
B. Sets the green component of each pixel to be the same as the blue component
C. Sets the blue component of each pixel to be the same as the green component
D. Loops over all pixels in pixelArray. For each pixelObj calls getGreen and stores that into value. Then sets value into blue.
E. None of the above.
What picture most accurately describes what this code does?

```java
Pixel[] pixelArray = this.getPixels();
int value = 0;
Pixel p = null;
for(int index = 0; index < pixelArray.length - 1; index++)
{
    p = pixelArray[index+1];
    q = pixelArray[index];
    p.setRed(q.getRed());
    p.setBlue(q.getRed());
    p.setGreen(q.getGreen());
}
```
World world1 = new World(200,100);
Turtle maria = new Turtle(25, 25, world1);

int numberOfIterations = 4;
int index = 0;
int value = 10;
while (index < numberOfIterations)
{
    maria.forward(value);
    value = value + 2;
    index = index + 1;
}

A. Creates 4 Turtles and moves them forward 12 units
B. Creates 1 Turtle and moves it forward 52 units
C. Creates 4 Turtles and moves them forward 52 units.
D. Creates 1 Turtle and moves it forward 12 units.

* This is something you WILL be asked to do on an exam/quiz (though not on Thursday). You should also be able to draw the memory model so do it now!
• Consider the following code:

```java
int value = 0;
Pixel[] pixArray = this.getPixels();
int index = 0;
Pixel p1 = pixArray[0];
value = p1.getRed();
```

What are the types of: `pixArray`, `p1`, `value`?

A. Pixel, Pixel, int
B. Pixel array, Pixel, int
C. Pixel array, Pixel, double
D. Pixel, double, int

(Which of these store references?)
Nested Loops: How do they work?
What order are pixels changed?

• A method in Picture.java… what does it print if width is 2 and height is 3?

```java
Pixel p;
for (int foo = 0; foo < getWidth(); foo++) {
    for (int bar = 0; bar < getHeight(); bar++) {
        System.out.println(foo + " " + bar);
    }
}
```

A. 0 0
    0 1
    1 0
    1 1
    2 0
    2 1

B. 0 0
    1 0
    2 0
    0 1
    1 1
    2 1

C. 0 0
    0 1
    0 2
    1 0
    1 1
    2 0

D. 0 0
    1 1
    2 2

1) Solo: (30 sec)  
2) Discuss: (2 min)  
3) Group: (30 sec)
Why does this have an error?

In a method in Picture.java... assume height=50, width=100

```java
Pixel p;
for (int bar = 0; bar < getWidth(); bar++)
{
    for (int foo = 0; foo < getHeight(); foo++)
    {
        p = getPixel(foo, bar);
        p.setColor(Color.BLACK);
    }
}
```

A. It doesn’t, this loops across rows, top to bottom
B. It doesn’t, this loops down columns, left to right
C. It tries to index a pixel off the end of a row (x value too big)
D. It tries to index a pixel off the end of a column (y value too big)
What are the first \((x,y)\) coords for \(\text{topP}\) and \(\text{bottomP}\) to mirror around horizontal axis?

\[
\begin{array}{cc}
\text{topP} & \text{bottomP} \\
A. & \\
(0,0) & (0,3) \\
(0,1) & (0,2) \\
(1,0) & (1,3) \\
B. & \\
(0,0) & (0,3) \\
(1,0) & (1,3) \\
(2,0) & (2,3) \\
C. & \text{either A or B will work} \\
D. & \text{none of the above}
\end{array}
\]
Pixel arrays and pixels in Pictures

1. How do the two relate to each other…?

```java
Picture pic = new Picture("mypic.jpg");
Pixel[] pixArray = pic.getPixels();
```

Given an x,y position in pic, what is a formula to calculate the position, i, of that Pixel in pixArray?

A. \( i = \text{pic.getWidth()} \times y + x \)

B. \( i = \text{pic.getHeight()} \times y + x \)

C. \( i = \text{pic.getWidth()} \times x + y \)

D. \( i = \text{pic.getHeight()} \times x + y \)

E. None of these
DEBUGGING: This code should swap the red and blue components at each Pixel; what does it ACTUALLY do?

Pixel[] pixelArray = this.getPixels();
int value = 0;
int index = 0;
while (index < pixelArray.length) {
    Pixel pix = pixelArray[index];
    value = pix.getRed();
    value = pix.getBlue();
    pix.setRed(value);
    pixelArray[index].setBlue(value);
    index++;
}

A. It has a compiler error
B. It sets the red value to be the same as blue
C. It sets the blue value to be the same as red
D. It really does swap them

How could we fix it?
In Picture.java...
public Picture copyRegionToNew(int xSource,
    int ySource, int xTarget, int yTarget )
{
    Picture newCanvas = new Picture();
    Pixel sPixel, tPixel = null;
    for (int sX = xSource, tX = xTarget; sX < 100+xSource; sX++, tX++)
    {
        for (int sY = ySource, tY = yTarget; sY < 100+ySource; sY++, tY++)
        {
            sPixel = this.getPixel(sX,sY);
            tPixel = newCanvas.getPixel(tX,tY);
            tPixel.setColor(sPixel.getColor());
        }
    }
}

In main...
Picture fish = new Picture( "fish.jpg" );
Picture newCanvas = fish.copyRegionToNew(10, 30, 50, 50);
newCanvas.show();

What error will the following code produce?
A. This code will not compile
B. The line “Picture newCanvas = fish.copyRegionToNew…” in main will cause an error
C. The line newCanvas.show() will cause an error
public void fillBottom(Color newColor)
{
    Pixel pix;
    for (int y = 0; y < this.getHeight(); y++)
    {
        for (int x = 0; x < this.getWidth(); x++)
        {
            <<<SELECT LINE OF CODE>>>>
        }
    }
}

A) if(y < this.getHeight()/2)  B) if(y > this.getHeight()/2)

C) if (this.getPixel(x, y) < this.getHeight()/2)  D) if (this.getPixel(x, y) > this.getHeight()/2)
public void everyOtherColumn(Color newColor) {
    Pixel pix;
    for (int y = 0; y < this.getHeight(); y++) {
        for (int x = 0; x < this.getWidth(); x++) {
            pix = this.getPixel(x, y);
            pix.setColor(newColor);
        }
    }
}

A) if(x<this.getWidth()/2)  B) if(x<this.getHeight()/2)
C) if ((x % 2) == 0)  D) if ( (this.getPixel(x,y) % 2) == 0)
CS Concept: Booleans are values

```c
if ( absValZ < 2.0 )
{
    return true;
}
else
{
    return false;
}
```

Which of the following is equivalent to the above code?

A. `return absValZ;`
B. `return absValZ < 2.0;`
C. `return absValZ >= 2.0;`
D. None of these
What's printed by this code? (assume calling object as shown)

A. 0,9
B. 60,0
C. 90,5
D. 100,4
E. None of the above
### Hiding information in images

<table>
<thead>
<tr>
<th>Red</th>
<th>Green</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>(39,</td>
<td>56,</td>
<td>101)</td>
</tr>
<tr>
<td>(37,</td>
<td>59,</td>
<td>100)</td>
</tr>
</tbody>
</table>

What is the maximum amount we can change a value (in decimal) by changing its three least significant digits?

A. 1  B. 3  C. 4  D. 7  E. 8

(00100111, 00111000, 01100101)

(00100101, 00111011, 01100100)
public Sound raiseP() {
    Sound highP = new Sound(this);
    SoundSample[] original = this.getSamples();
    SoundSample[] higher = highP.getSamples();
    int newPlace = 0;
    for (int origI = 0; origI < original.length; origI+=2) {
        
    }
How would the code below change the SoundSample array?

// In the sound class
public void mystery() {
    SoundSample[] original = this.getSamples();
    for ( int index = 0; index < original.length; index++ ) {
        original[index].setValue( original[index/2].getValue() );
    }
}

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>150</td>
<td>200</td>
<td>300</td>
<td>140</td>
<td>10</td>
<td>-40</td>
<td>-100</td>
<td>-250</td>
<td>-150</td>
</tr>
</tbody>
</table>

A | B | C | D
---|---|---|---
| 100 | 100 | 150 | 150 | 200 | 200 | 300 | 300 | 140 | 140 |
| 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 100 | 200 | 140 | -40 | -250 | 10 | -40 | -100 | -250 | -150 |

It causes an error

DRAW!!
int x = 42;
int y = x + 1;
x = y + 1;
System.out.println( x + " , " + y );
public void silly(int a, int b) {
    a = b + 1;
    b = a/2;
    System.out.println( a + " , " + b);
}

// in main
int a = 67;
int b = 13;
silly( b, a );

A. 67, 13
B. 68, 34
C. 14, 7
D. 8, 7
E. Something else
What is printed? (Draw boxes and scope!)

```java
public void silly( int a, int b ) {
    a = b + 1;
    b = a/2;
}

// in main
int a = 67;
int b = 13;
silly( b, a );
System.out.println( a + "", " + b );
```

A. 67, 13  
B. 68, 34  
C. 14, 7  
D. 8, 7  
E. Something else
public void silly( int a, int b ) {
    a = b + 1;
    b = a/2;
}

// in main
int a = 67;
int b = 13;
silly( b, a );
System.out.println( a + ", " + b );

A. 67, 13
B. 68, 34
C. 14, 7
D. 8, 7
E. Something else

Different a’s and b’s!
Reassignment within the function has NO EFFECT on the variables in the interaction pane.
public void doStuff() {
    pic = new Picture( this.getWidth(), this.getHeight() )
    for ( int x=0; x < pic.getWidth(); x++ ) {
        for (int y=0; y < pic.getHeight(); y++ ) {
            Pixel p = pic.getPixel( x, y );
            p.setColor( new Color( 100, 100, 100 ) );
        }
    }
}

// in main
Picture myP = new Picture(FileChooser.pickAFile() );
myP.doStuff();
myP.show();

Assume that the user chose a butterfly picture when the dialog box came up. What picture will be displayed?
A. A butterfly
B. A picture that is all gray
C. A gray-colored butterfly
D. Something else
public void doStuff() {
    pic = new Picture( this.getWidth(), this.getHeight() )
    for ( int x=0; x < pic.getWidth(); x++ ) {
        for (int y=0; y < pic.getHeight(); y++ ) {
            Pixel p = pic.getPixel( x, y );
            p.setColor( new Color( 100, 100, 100 ) );
        }
    }
}

// in main
Picture pic = new Picture( FileChooser.pickAFile() );
pic.doStuff( );
pic.show();

Assume that the user chose a butterfly picture when the dialog box came up. What picture will be displayed?
A. A butterfly
B. A picture that is all gray
C. A gray-colored butterfly
D. Something else
public Picture doStuff() {
    pic = new Picture( this.getWidth(), this.getHeight() )
    for ( int x=0; x < pic.getWidth(); x++ ) {
        for (int y=0; y < pic.getHeight(); y++ ) {
            Pixel p = pic.getPixel( x, y );
            p.setColor( new Color( 100, 100, 100 ) );
        }
    }
    return pic;
}

// in main
Picture pic = new Picture( FileChooser.pickAFile() );
pic.doStuff();
pic.show();

Assume that the user chose a butterfly picture when the dialog box came up. What picture will be displayed?
A. A butterfly
B. A picture that is all gray
C. A gray-colored butterfly
D. Something else
public Picture doStuff() {
    pic = new Picture( this.getWidth(), this.getHeight() )
    for ( int x=0; x < pic.getWidth(); x++ ) {
        for (int y=0; y < pic.getHeight(); y++ ) {
            Pixel p = pic.getPixel( x, y );
            p.setColor( new Color( 100, 100, 100 ) );
        }
    }
    return pic;
}

// in main
Picture pic = new Picture( FileChooser.pickAFile() );
pic = pic.doStuff();
pic.show();

Assume that the user chose a butterfly picture when the dialog box came up.
What picture will be displayed?
A. A butterfly
B. A picture that is all gray
C. A gray-colored butterfly
D. Something else
public class Point
{
    private double x;
    private double y;

    public Point(double x_in, double y_in)
    {
        this.x = x_in;
        this.y = y_in;
    }

    public static void main( String[] args )
    {
        double d = 42.0;
        Point p;
        p = new Point(d, 42.0);

        Point q = new Point(p.x, 42)
    }
}
public class Point
{
    private double x;
    private double y;

    public Point(double x_in, double y_in)
    {
        this.x = x_in;
        this.y = y_in;
    }

    public static void main( String[] args )
    {
        double d = 42.0;
        Point p;
        p = new Point(d, 42.0);

        Point q = new Point(p.x, 42)
        d = 65.0;
        p.x = 55.0;
        System.out.println( d + "", "", + p.x + "", "" + q.x )
    }
}
Challenge: Create a checkerboard

Write a method that will copy a pattern of 20x20 squares from a source image to a new Picture, and return the new picture.
Exam 3 practice problem

• Write a method in the Picture class that draws a 10x10 red square in the center of the calling object.

• Do this both by looping over the whole image and using an if statement, and by looping over ONLY the specified region.
Exam 3 practice problem

• Write a method in the Picture class that takes a threshold value for green (an int). It then creates a new Picture and copies into this new Picture only those pixels whose green value is above the threshold. It leaves the rest of the Pixels in the new Pictures blank. It returns the new Picture.
Exam 3 practice problem

• Write a method in the Picture class that takes a target Picture object and copies the calling object’s picture upside down onto the target picture. Can you handle the case where the target is smaller than then calling object? How about where the target is larger?
• Study
• CAPES and Supplemental survey (if you haven’t already)