Big collection of midterm review questions. Email Prof. Chaiken before 3:00PM Tues, Mar 6, with suggestions of which to review in class before the exam.

When there are choices, always select the BEST from what's given, even if one might argue for a worse choice. Answers are graded on how well they indicate your understanding of the subject content.

1. Which is not a method of the Picture class?
   a. `getPixels()`    
   b. `getWidth()`     
   c. `getHeight()`    
   d. `getColor()`     
   e. `Picture(String)`

2. What would be the result of 9/2?
   a. 4
   b. 4.5
   c. 5
   d. 1
   e. The answer is undefined.

3. What would be the result of 9.0/2?
   a. 4
   b. 4.5
   c. 5
   d. 1
   e. The answer is undefined.

4. What would be the result of ((int) (9.0/2.0))?
   a. 4
   b. 4.5
   c. 5
   d. 1
   e. The answer is undefined.

5. Which flowchart box is used to decide what to do based on a true/false test?
   a.  
   b.  
   c.  
   d.  

6. Which flowchart box is used to hold some computing instructions?
   a.  
   b.  
   c.  
   d.  

7. Which combination of (red, green, blue) values might show up as orange?
   a. (180, 120, 50)
   b. (120, 180, 180)
   c. (120, 120, 180)
   d. (180, 50, 120)
   e. (50, 120, 180)

8. What is the body of a for loop?
   a. The whole loop.
   b. The part done first and only once.
   c. It specifies the test to continue looping.
   d. The nested onion ring of code just after for ( ??; ??; ??)
   e. The part that frequently looks like x = x + 1

9. Which series of statements could have produced the Turtle path drawn on the right, assuming the Turtle started at A facing north (up)?
   a. forward();
      turnRight( );
      forward( );
      turn(90);
      forward(50);
   b. forward();
      turn(90);
      forward( );
      turn(90);
      forward(50);
   c. forward();
      turnRight( );
      forward( );
      turn(90);
      forward( );
   d. forward();
      turn(-90);
      forward( );
      turn(90);
      forward(50);
   e. forward();
      turnRight( );
      forward( );
      turnRight( );
      forward(50);
10. What are the (x,y) coordinates of the Pixel in the upper right corner of the screen? Assume the Picture has a height of 640 and width of 480
   a. (0,0)
   b. (0,639)
   c. (0,479)
   d. (639,0)
   e. (639,479)

11. What are the (x,y) coordinates of the pixel in the upper right corner of the screen? Assume the Picture has a height of 640 and width of 480
   a. (0,0)
   b. (0,480)
   c. (0,479)
   d. (639,0)
   e. (640,0)

12. Besides having potential behaviors named by method names and coded by method bodies, what else might a Java object (like a Turtle, Picture or Pixel) have?
   a. A computer.
   b. Priority.
   c. Numbers or other data written on it, like if the object were a ticket.
   d. Physical mass and chemical composition, like 12 kilograms of iron.

13. Which example of code illustrates a caller?
   a. Turtle mT = new Turtle( new World ( ) );
      mt.drawS( 3.4 );
   b. void drawS( double scale )
      {
         int temp;
         temp = (int)( scale*3.5 );
      }

14. Which example of code illustrates a callee?
   a. Turtle mT = new Turtle( new World ( ) );
      mt.drawS( 3.4 );
   b. void drawS( double scale )
      {
         int temp;
         temp = (int)( scale*3.5 );
      }

15. What does the caller do?
   a. It writes parameter values and perhaps an object location on a calling card, makes the call, and waits for the callee to do something and return.
   b. It reads parameter values and perhaps an object location from a calling card, does something for the caller, and tells the caller it's done by returning.
16. What does the callee do?
   a. It writes parameter values and perhaps an object location on a calling card, makes the call, and waits for the callee to do something and return.
   b. It reads parameter values and perhaps an object location from a calling card, does something for the caller, and tells the caller it's done by returning.

17. If we think of one Pixel object as a ticket, what data is written on it?
   a. Location (x,y coordinates) in its Picture, only.
   b. Shape
   c. Color, written as 3 RGB numbers, only.
   d. Location (x,y coordinates) in its Picture and Color (RGB numbers)
   e. Priority

18. Briefly explain how, when you are doing project work, you would add a new method, say to draw a letter, to the Turtle class?

19. How would you give that method parameters?

20. How would you code a call to that method within application code written in main()? 

21. How would you code a call to that method as above, so that you specify parameter values in the call?

22. Explain how a computer writes and reads parameter values in terms of a person being a caller, a person being a callee, and the caller preparing a calling card he or she gives to the callee.

23. Calculate on paper the average of the two numbers 9.0 and 12.0; show why it is 10.0

24. Calculate on paper the weighted average of
   9.0 with weight 1/3
   combined with
   12.0 with weight 2/3

   Show why the answer is 11.0

Consider an image with a height of 50 and width of 100 pixels. What is the index of the top right-most pixel in the one dimensional array returned by picObj.getPixels()? 

Write the single integer answer below: 

________________________

What is the range of integer values that can be red intensities of a Color?
Minimum____________________ Maximum__________________

The greater the intensity, the
(A) Brighter
(B) Darker
does the Color appear.
Consider the sequence of Java statements:
```java
double a = 2.5;
int b = 6;
int c = 4;
int d = 8;
int x;
```

What is printed when each of the following statements is inserted into the space indicated above?

A) `x = b + d * x;`  
B) `x = b + d / c;`  
C) `x = (int) (c / a);`  
D) `x = d % 6;`  
E) `x = (int) a * b;`  

Write a brief description of what the following terms mean:

25. Pixel
26. Declaration
27. Object
28. Array
29. The operation symbolized by `=` (single equal sign) in Java.

Filling in code
For each of the following questions, fill in the missing Java statement in the box provided:

30. The following code finds the sum of all of the items in the integer array named `myArray`:
```java
int sum = 0;
for (int value : myArray) {
    // Add the missing statement here to calculate the sum.
}
```

31. The following code reduces the amount of blue in the Picture object `pict`:
```java
Pixel [] pixels = pict.getPixels();

// Make a loop here whose body includes the 2 lines below.
```
```java
pix.setBlue(pix.getBlue()-30);
```
32. The following method `ave(int, int, int)` calculates and prints the average of three numbers provided as parameters:

```java
public void printAve(int num1, int num2, int num3) {
    double average;
    average = (double)(num1 + num2 + num3) / 3.0;
}
```

33. Write the series of Java statements (they do not have to be contained within a method) to create a `Picture` object using the file located at `c:/System/Bitmaps/castle.bmp` and demonstrate how a loop can be coded to count the number of pixels in that picture. Declare all necessary variables. (Hint: Code `count=count+1` to be done for each Pixel in the Picture.)

34. Explain a better way to solve this by multiplying, WITHOUT A Loop!

35. The code below contains a `main` method with some missing code. Put Java statements in the space below to perform the following: Create a `World` and create two `Turtles` in that world. These two `Turtles` should draw a square, such that the length of each side is 200. Each `Turtle` must travel the same distance - that is, each `Turtle` must draw exactly half the square. Hint: remember that all `Turtles` begin in the center of the World, facing North or up.

```java
public class TwoTurtleSquare {
    public static void main(String[] args) {
        // Your code begins here
        ..more space is given in a real exam...
    }
}
```
36. Code-tracing is especially handy for understanding the logic behind a given piece of code. Consider the follow code segment:

```java
int index = 1;
int count = 0;
for ( /* nothing*/ ; index <= 3 ; /* nothing */) {
    count = index;
    index = index + 1;
}
System.out.println(count);
```

What is printed?(just one number) _____________________________

Use the following boxes to simulate the tickets (i.e., variables) used by a computer who (or which) follows directions expressed by the Java code. Keep track of the number currently on each ticket (a) after the first two lines of code, and then, (b) during each and every repetition of the code in the for loop's body. Remember to heed the names (index and count) of the two tickets.

Hint: There might be more boxes than you need to use because the computation stops before you use them up. It's your job to figure out when the loop stops!

<table>
<thead>
<tr>
<th>index</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WARNING: Now that you simulated the computation, GO BACK and CHECK and probably FIX UP your answer to what is printed!

Simulating unfamiliar code is tricky and most beginners will get the prediction of what's printed WRONG before they do the full simulation! (One college exams, it's wise to go back and check earlier question subparts after reaching the end of a big problem.)
QUESTION 2 (18 points)

The following code is a method within Picture.java. It contains six errors that will prevent the code from compiling and/or running. Circle the location of each error, then below each line that contains an error write an error-free version of that line.

```java
public void changeImage(Color)
{

    for(int x = 0; x < getWidth(); x++)
    {

        for(int y = 0; y < this.getHeight() y++)
        {

            Pixel pixel = this.getPixels(x,y);

            Color color = pixl.getColor();

            pixel = setColor( Color.blue );

        }
    }
}
```
QUESTION 6 (7 points)

Consider the following method in the Picture class:

```java
public void modifyPicture()
{
    Pixel pixel = null;
    Pixel[] pixels = this.getPixels();
    for(int x = 0; x < this.getPixels().length; x++) {
        pixel = pixels[x];
        int r = pixel.getRed()/2;
        int g = pixel.getGreen()/2;
        int b = pixel.getBlue()/2;
        pixel.setColor(new Color(r, g, b));
    }
}
```

Describe the effect of this method on a picture in a few sentences:

Show the effect of this method by drawing the result in the "After" frame:

Before

After