

The University of Western Ontario
CS1026a - Summer 2015
Assignment 3

DUE: Thursday, June 11th, 11:55pm
Weight: 15%

Purpose:

To gain experience with:

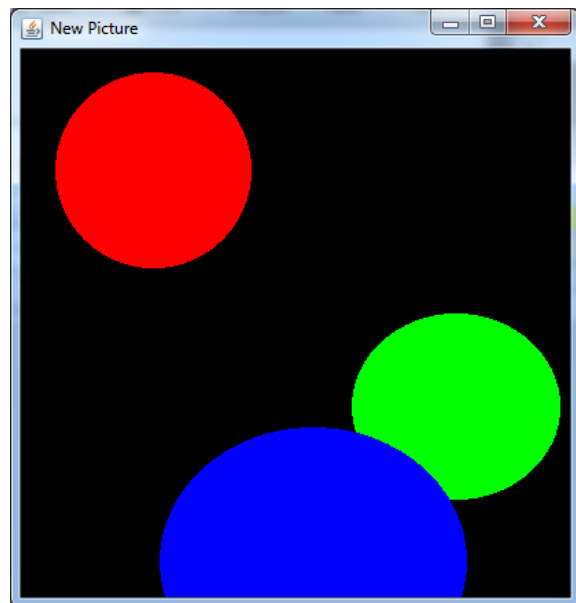
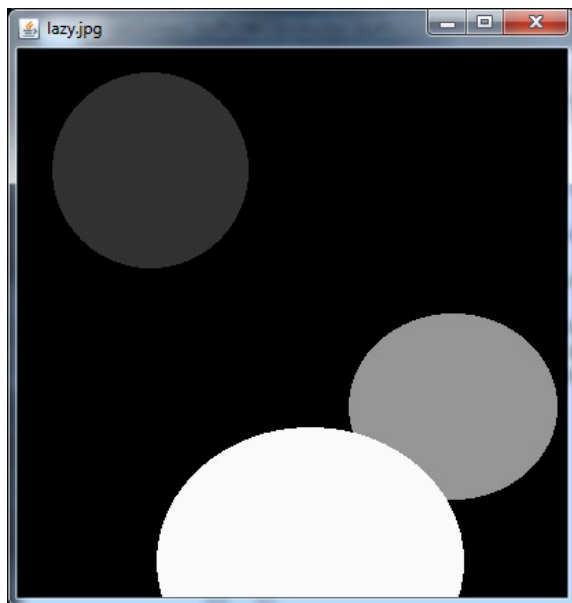
- Using methods and returning values
- Using nested loops
- Picture manipulation
- Conditionals

Part A:

In the assignment file you will see three beautiful pictures painted by Snow White's three lesser known (and less desirable) dwarf friends. Unfortunately they didn't pay attention and saved their files as JPEGs. This means that their pictures have been saved in a format that loses quality and their paintings are not as clear as one would hope (open the files in a regular picture viewer and zoom in, notice how the image gets 'grainy'?). Additionally, they also seem to be obsessed with the colour grey.

Our goal is to improve the quality of their pictures and colour things up a bit.

Write a picture object method to create and return a new Picture. This method will fix the very minor graininess and change the grey colours to brighter colours.



The method must be defined as follows:

```
public Picture correctColours(Color c1, Color c2, Color c3)
```

The three colour parameters will be the colours to change the darkest, middle, and brightest greys too (respectively). In the above example the colours `java.awt.Color.RED`, `java.awt.Color.GREEN`, and `java.awt.Colot.BLUE` were used. **Also notice the return type, it's not `void`, it's a `Picture`.**

This method will:

1. Create a new picture
2. Iterate through every pixel
 - a. If the grey is between 25 and 100, make the corresponding pixel in the new picture `c1`
 - b. If the grey is between 100 and 200, make the corresponding pixel in the new picture `c2`
 - c. If the grey is above 200, make the corresponding pixel in the new picture `c3`.
 - d. Otherwise, make the corresponding pixel `BLACK`
3. Return the new picture

Call this method from the main method from within a Class you create titled "Assign3". See below for some help. You can pass whichever colour you would like to the newly created picture object method (you may need to look at the java documentation if you want to get creative:

<https://docs.oracle.com/javase/7/docs/api/java/awt/Color.html>).

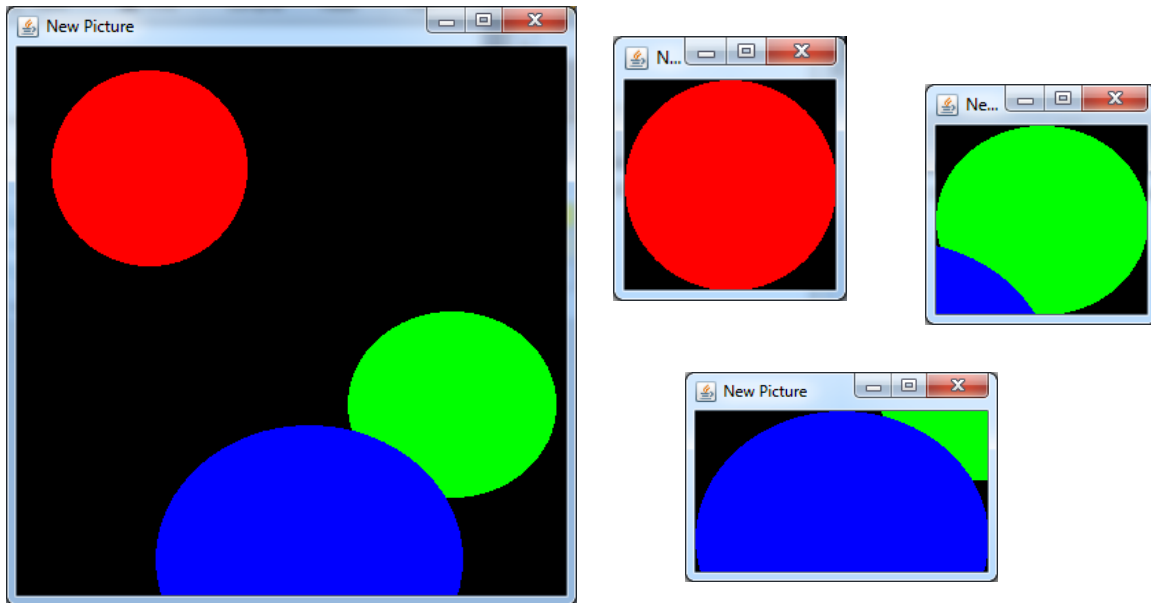
```
public class Assign3
{
    public static void main(String[] args)
    {
        Picture pic1 = new Picture("lazy.jpg");
        pic1.show();
        Picture pic2 = pic1.correctColours(java.awt.Color.RED,
            java.awt.Color.GREEN, java.awt.Color.BLUE);
        pic2.show();
    }
}
```

Just like assignment 2, your pictures must be opened from the working directory. You cannot use a `FileChooser`.

Part B:

Prince Charming has seen the coloured versions of the dwarf's paintings and thinks that they are exceptionally beautiful; however, he feels that he cannot fully appreciate the exquisiteness unless the masterpieces are broken down into their components.

Our new goal is to write a series of new object methods to break the new picture down into even more new Pictures containing the components of the colouredized image.

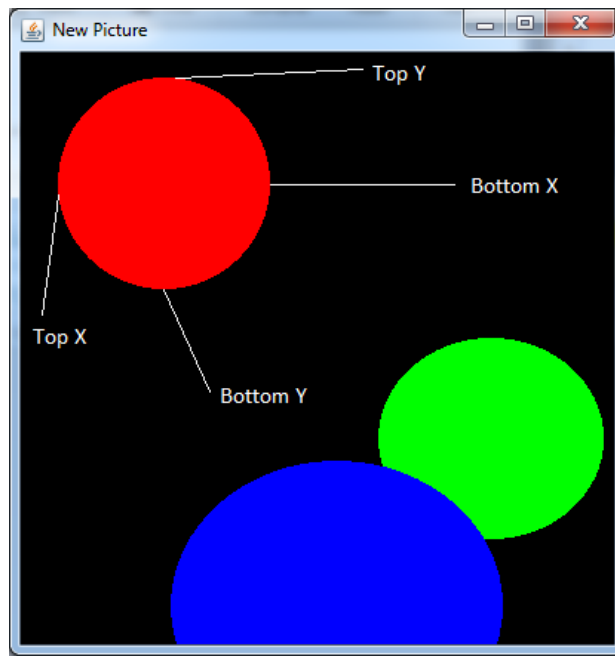


Five methods are required to do part B. The first four are as follows (take note that these methods are `private`, not `public`), and must be defined as written below:

```
private int topX(Color c)
private int topY(Color c)
private int bottomX(Color c)
private int bottomY(Color c)
```

These methods will scan the picture object that invoked the method to find the desired pixel matching the colour passed to the method as a parameter. Let's take `topY(Color c)` for example. This method will scan the picture until it finds the first pixel in the Y direction that has the same colour as `c`. Refer to the following image for visualization.

REMEMBER: To compare objects we need to use an object method which returns a Boolean called `.equals(Object o)`; we cannot compare objects with `"=="`. What this means is that when comparing colours we must type `"someColour.equals(c)"`; we cannot type `"someColour == c"`.



The last new method to be written will do the actual cropping and return the cropped image. This method must be defined as follows:

```
public Picture cropPicture(Color c)
```

This method will create a new picture with the proper size, crop the shape from the invoking picture with the corresponding `Color c` (the parameter passed), and return the new picture with the cropped image (like the images above). This method is required to call the four `private` methods defined earlier.

This method will be invoked three times from the main method. Add the following code to your main method and adjust it work with the same three colours you used for Part A.

```
Picture redP = pic2.cropPicture(java.awt.Color.RED);  
Picture greenP = pic2.cropPicture(java.awt.Color.GREEN);  
Picture blueP = pic2.cropPicture(java.awt.Color.BLUE);  
redP.show();  
greenP.show();  
blueP.show();
```

Test your program on all thee of the dwarf's paintings. Notice anything strange about Meanie's picture?

You do not need to save the output images.

Non-functional Specifications:

1. Include comments for your classes

```
// CS1026B Assignment 3
// Your name
// A brief description of what this program does
```
2. Include brief comments in your code that explain what each method is doing.
3. Include comments for any code which is unclear.
4. Assignments are to be done individually and must be your own work. Software will be used to detect cheating.
5. Use Java conventions and good Java programming techniques, for example:
 - i. Meaningful variable names
 - ii. Conventions for naming variables and constants
 - iii. Use of constants where appropriate
 - iv. Readability: indentation, white space, consistency

Important! You must follow all the specifications above for your program

What You Will Be Marked On:

1. Functional specifications:
 - ✓ Are the required methods written according to specifications?
 - ✓ Do they work as specified? Pay attention to details!
 - ✓ Are they called as specified?
 - ✓ Are the two objects drawn according to the specifications?
2. Non-functional specifications: as described above

What to Submit:

*Submit .java documents. Be VERY careful **not** to submit the .java~ version or the .class file. This can be tricky, as your system may not show the file extension in the file selection dialog.*

1. *Picture.java* (with your **new methods**)
2. *Assign3.java*
3. The original JPEG files
 - meanie.jpg
 - lazy.jpg
 - smelly.jpg

How to Submit:

You are to submit the specified files through the online OWL submission for Assignment 3 under the “Assignments” tab.