

CS1026a: Assignment 2

Due: Wednesday, October 30th, 9pm

Weight: 8%

Purpose:

To gain experience with

- Loops in Java (for loops and while loops)
- Picture manipulation

Task:

Your task is to create a collage of various images. A collage is “an artistic composition of various items glued onto a surface”.

Part A:

In this part of the assignment, you will add two **new** methods to the Picture class, each of which does some kind of picture modifications. You can modify the pictures in any way you like! Check out the textbook and lectures for some examples. Be creative – you can decrease the color, add lines, make black and white, blur, etc.

Functional Specifications for Part A:

1. One of your new methods must use a **for loop** and one must use a **while loop**.
2. You should add your two new methods to the Picture class, after the line that looks like this:

//////////////////////////////// methods //////////////////////////////////

Add them to the file *Picture.java* that you downloaded and used in Lab 5.

3. To show that your two new methods work, you will write a Java test program in a class you make called TestMethods that will be saved as *TestMethods.java*. The code to get you started on TestMethods is provided below; you need to add statements that a) create picture objects and b) invoke your methods on them to change the pictures and show your methods work. Your test program must include code that **displays** both the **original image** and the image **after** it has been changed by your method. Here is an algorithm for the test program:

- Get an original image (use FileChooser)
- Display it on the screen
- Change it using your first method
- Display the **changed** image on the screen
- Get an original image (use FileChooser)
- Display it on the screen
- Change it using your second method
- Display the **changed** image on the screen

Code to start with:

```
import java.awt.Color;
public class TestMethods
{
    public static void main(String[] args)
    {
        /* Insert your code here */
    }
}
```

4. You must use one (or two) of the images provided in TestImages.zip to test your methods. You may use the same image to test both methods if you like. You can download TestImages.zip from the course webpage in the Assignment 2 folder and save it in your home directory. You will need to extract the files.

Part B:

Make a collage! In this part of the assignment you will create a collage of images on the computer using the methods you have created. You will take images, use your methods on them, and then copy them onto a blank canvas as described below. You can be as creative as you want, and you may use the provided images or some of your own!

There will prizes for creativity – so do your best!

Functional Specifications for Part B:

1. Your collage will consist of 5 images:
 - i. An original image (your choice).
 - ii. Two images that are modifications of the original, modified using your two methods from Part A.
 - iii. Two images that are modifications of the original, modified using any methods found in the lecture notes, labs or textbook.
2. How you arrange the images in the collage is entirely up to you.
3. Since your original image can be one of the ones provided or your own, you may need to reduce the size of an image that you want to use. This can be done using commercial software or the method `getPictureWithHeight` in the picture class. This method will create a picture with the specified height from an existing picture, and will keep the same height/width ratio. Here is an example of how to use the method:

```
/* assume you are starting with a Picture object referenced by largerPic */
Picture smallerPic = largerPic.getPictureWithHeight(200);
smallerPic.write("filepath goes here /smallerPic.jpg");
```

4. Your program must write your collage image to a file named myCollage.jpg

Non-functional Specifications:

1. Include brief comments in your code that explain what each method is doing.
2. Assignments are to be done individually and must be your own work. Software will be used to detect cheating.
3. Use Java conventions and good Java programming techniques, for example:
 - i. Meaningful variable names
 - ii. Conventions for naming variables and constants
 - iii. Readability: indentation, white space, consistency

What to Hand In:

1. *Picture.java* (with your **new methods**), *TestMethods.java*, *MakeCollage.java*
2. The JPEG file with the original image used for creating your collage
3. The file *myCollage.jpg*

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
3. Assignment submission: via the OWL, though the assignment submission in OWL.