

# HW1: AdvancedPicture Methods

DUE DATE: Friday January 18, 2008

For this assignment, modify the **AdvancedPicture** class that extends **Picture** and write two new methods in **AdvancedPicture** that implement two different kinds of image manipulation.

## Image Manipulation

The type of image manipulation implemented is entirely up to you. However, you must implement a manipulation that does not already exist within the **Picture** class. You are more than welcome to use the methods within **Picture** as a guide, but we expect you to write your own methods and to not make any calls to existing **Picture** methods.

## Self-Documentation

Suppose we have two methods named **intensifyColors()** and **method1()**. Can you guess as to what each method will do without testing the method or reading through the code? **intensifyColors()** seems like it will make the colors brighter or sharper somehow, while **method1()** could do anything. Now imagine a giant program where all the methods were named similar to **method1()**. Get the big picture now? Your methods should really have names that reflect the manipulations that will occur. Giving your methods names that describes what happens within the method is called self- documentation. Self-documentation makes it easier for others to understand what the method is supposed to do and the grand scheme of things what your program will do.

## Commenting

Though your method names are self-documenting you should still comment your code. Place above each of your method names, comments about what the method will do. Also always place comments that include your name, prism ID (gtx000x or jsmith3) and grading TA at the top of all the java files you submit. Some people when asked to comment their code will do so after all of the code is already written and working. While this is an acceptable way of commenting, some people find it easier to write an outline of how you want to approach a program using comments. Essentially we use write a program to achieve some goal or complete some task. While some people can start hacking their way into some semi-functioning code without much forethought, it is highly recommended that you actually sit and think about the steps you should take. Regardless of when you comment, you should comment and will be required to comment in this homework and forthcoming homeworks.

## Looping

Also you will be required to use a **for** loop in one method and a **while** loop in the other. The method you write will be called on the picture that needs to be manipulated. For instance, after this sequence of code:

```
AdvancePicture ap =
    new AdvancedPicture(FileChooser.getMediaPath("swan.jpg"));
ap.intensifyColors();
ap.show();
```

you would expect `ap` to be manipulated in some way.

## What to Return from the Methods

You are only required to write `void` methods, methods that do not return anything after it is performed. If you so desire, you may write non-void methods, methods that does return something after completion, but know that it is not required for this assignment.

## Testing the Methods

Remember that you were first introduced to the `main` method in homework 0 where your `HWO` class contained a `public static void main(String[] args)`. You should also include a `main` method within `AdvancedPicture` to test your new picture manipulation methods. Remember to test both methods preferably on a fresh picture each time.

You may not include `FileChooser.setMediaPath()` anywhere in your code. You must use `FileChooser.getMediaPath()` and may not hardcode any picture paths in your homework.

## What to Turn In

- `AdvancedPicture.java`

## How to Turn In

- Turn in via [TSquare](#).