Making a color spot on (a picture of) somebody's nose

Step 1 Make a directory (folder) for this lab work Lab5

In the Lab (or other Linux/Unix computer), reminder
1. Right after logging in you will be in your home directory.
2. cd CS1201
3. mkdir Lab5
4. cd Lab5

On a computer with a GUI shell
1. From MyComputer or wherever you start at, navigate to and into your CS1201 folder (directory).
2. Request New … Folder
3. Name or rename the New Folder as Lab5

Step 2
Get a digital portrait, like http://www.cs.albany.edu/~sdc/CSI201/sdc.JPG into your Lab05 directory (stop saying folder)

For speed, use the TA's computer.
Go to http://www.cs.albany.edu/~sdc/CSI21/Spr14/PictureGetting/GetPicture.html
Use username cs201 and password admin201
outside the lab, you can right-click your photo to download it. It’s wise to right away download it into the Lab05 directory you had just made.

In the Lab: (1) Make sure you’re in your Lab5 directory with pwd for print working directory. (2) Noting there is space AND ONE DOT (period) at the end, give the copy command below:
cp /usr/local/depts/cs/geintro/Fall13Photos/<your NetID>.JPG

This means put in your own UA NetId, two lower case letters followed by some digits 0–9, between the / and the .JPG. So, if your NetId were ab214365 (which it isn’t), you would type
cp /usr/local/depts/cs/geintro/Spring13Photos/ab214365.JPG

Step 3
Write a Java application like

```java
public class Lab5App {
    public static void main(String[] a) {
        FileChooser.pickMediaPath();
        EditablePicture pRef;
        pRef = new EditablePicture(FileChooser.pickAFile());
        pRef.explore();
        java.awt.Color nc = ColorChooser.pickAColor();
        java.util.Scanner sc = new java.util.Scanner(System.in);
        System.out.printf("Type in x y coords of the spot.");
        int xIn;
        xIn = sc.nextInt();
        int yIn;
        yIn = sc.nextInt();
        pRef.changeXYLocationsColor( xIn, yIn, nc );
        pRef.explore();
    }
}
```

Save, TRY TO Compile, (correct typing mistakes) but it will fail!

Step 4
Extend the Picture class like

```java
public class EditablePicture extends Picture {
    public EditablePicture(String fn) { super(fn); }
    public void changeXYLocationsColor( int xPar, int yPar, java.awt.Color cPar ) {
        Pixel pixRef;//Set up a reference variable.
        pixRef = this.getPixel( xPar, yPar );//Make it refer
        pixRef.setColor( cPar ); //Actually change that
        System.out.println("");    
    }
}
```

/* The main below is an optional good idea. */
public static void main(String[] a)
{
    System.out.println("You’re using this software the wrong way!");
    Save, Compile, (correct typing mistakes).
    NOW, try to compile Lab5App, fix mistakes, and RUN Lab5App. BUT do Step 5 now.

Step 5
FileChooser.pickMediaPath();
makes a dialog that ONLY shows FOLDERS.
You will NOT SEE any names of files (NO .java, .JPG (digital images) or any other files).
The purpose is for you to pick the Lab5 folder/dir,
so the software will get back there when it runs again and again. That will save you time.

FIRST: Get to Lab5 and click Open!
Step 6 after clicking **Open**
An Explorer windows will show.
Pick a spot and note the X: Y: location coordinates

BookClass bug workaround: On some people’s systems, `explore()` fails. If it fails for you, replace the two
`pRef.explore( )`; method calls with `pref.show( )`;
It is almost as good. You will not see the Pixel locations
however.

Step 7

You should see the Color Chooser window.
Choose a new color for the Pixel you have located by x and y coordinates.

Step 8
Two G&E Explorer windows will show.
Slide one over so you can see both.

Step 9
In each Explorer, use the **Zoom** menu to zoom to 500%, the max.
Look for the changed pixel in second explorer to pop up!Find out the coordinates of a Pixel near the middle of the nose.
How?
Reset the Zoom back to 100% so you can see the nose.
Click on the middle of the nose.
Zoom in and shift the view to see the nose, close up.

(It's not ukky; it's just a picture!)

---

**x and y coords. locating the selected Pixel.**

- red, green and blue intensity levels (numbers) that make up the color information of the selected Pixel.
- A box filled with the color of the selected Pixel (flesh colored!)
- The yellow cross-hair locates the selected Pixel