Instructions:

E1: Steganography

Steganography is the science of hiding information in a picture. You can hide a black and white message inside a color picture by first changing all the red values in the original color picture to be an even value (by subtracting one if odd). Make a picture of the same size out of the message that will be hidden. Then loop through both the original picture and the message picture, setting the red value of a pixel in the original picture to odd (by adding one to it) if the corresponding pixel in the message picture is close to the color black. Write an encode method that takes the black and white picture message and changes the current picture to hide the message picture inside of it. Then also write a decode method that returns the picture hidden in the current picture. There is starter code for encode and decode in the Picture.java class below

Figure 1: original (left), message (middle), beach with message hidden (right)

E2: Chromakey

Write a chromakey method that replaces the current pixel color with the color from another picture at the same row and column when the current pixel color is close to a specified color. In many movies, the actors are filmed in front of a green screen and then the green is replaced with a different background using a similar technique. There is starter code for the chromakey method below.

The picture in Figure 2 is of Dr. Mark Guzdial of Georgia Tech. Dr. Guzdial is the creator of the Media Computation approach to teaching computing concepts, which has students write programs that manipulate media: pictures, sounds, text, and movies. These labs are based on his work.

Figure 2: Dr. Guzdial (left), moon (middle), Dr. Guzdial on the moon (right)

Code starters:

```
/** Method to replace the blue background with
  * the pixels in the newBack picture
  * @param newBack the picture to copy from
  */
public void chromakey(Picture newBack)
{
  Pixel fromPixel = null;
  Pixel toPixel = null;
  Pixel[][] toPixels = this.getPixels2D();
  Pixel[][] fromPixels = newBack.getPixels2D();
  . . . your code completed . . .
}


/** Hide a black and white message in the current
  * picture by changing the red to even and then
  * setting it to odd if the message pixel is black
  * @param messagePict the picture with a message
  */
public void encode(Picture messagePict)
{

  Pixel[][] messagePixels = messagePict.getPixels2D();
  Pixel[][] currPixels = this.getPixels2D();
  Pixel currPixel = null;
  Pixel messagePixel = null;
  . . . your code completed . . .
}

/**
 * Method to decode a message hidden in the
 * red value of the current picture
 * @return the picture with the hidden message
 */
public Picture decode()
{
  Pixel[][] pixels = this.getPixels2D();
  int height = this.getHeight();
  int width = this.getWidth();
  Pixel currPixel = null;
  Pixel messagePixel = null;
  Picture messagePicture = new Picture(height,width);
  Pixel[][] messagePixels = messagePicture.getPixels2D();
  . . . your code completed . . .
  return messagePicture;
}
```