JAVA DIP - GRAYSCALE CONVERSION

In order to convert a color image to Grayscale image, you need to read pixels or data of the image using **File** and **ImageIO** objects, and store the image in **BufferedImage** object. Its syntax is given below:

```
File input = new File("digital_image_processing.jpg");
BufferedImage image = ImageIO.read(input);
```

Further, get the pixel value using method **getRGB** and perform GrayScale method on it. The method getRGB takes row and column index as parameter.

```
Color c = new Color(image.getRGB(j, i));
int red = (c.getRed() * 0.299);
int green =(c.getGreen() * 0.587);
int blue = (c.getBlue() *0.114);
```

Apart from these three methods, there are other methods available in the Color class as described briefly:

Sr.No.	Method
1	**brighter** It creates a new Color that is a brighter version of this Color.
2	**darker** It creates a new Color that is a darker version of this Color.
3	**getAlpha** It returns the alpha component in the range 0-255.
4	**getHSBColor***floath, floats, floatb* It creates a Color object based on the specified values for the HSB color model.
5	**HSBtoRGB***floathue, floatsaturation, floatbrightness* It converts the components of a color, as specified by the HSB model, to an equivalent set of values for the default RGB model.
6	**toString** It returns a string representation of this Color.

The last step is to add all these three values and set it again to the corresponding pixel value. Its syntax is given below:

```
int sum = red+green+blue;
Color newColor = new Color(sum,sum,sum);
image.setRGB(j,i,newColor.getRGB());
```

Example

The following example demonstrates the use of Java BufferedImage class that converts an image to Grayscale:

```
import java.awt.*;
import java.awt.image.BufferedImage;

import java.io.*;

import javax.imageio.ImageIO;
import javax.swing.JFrame;

public class GrayScale {

   BufferedImage  image;
   int width;
   int height;

   public GrayScale() {

      try {
         File input = new File("digital_image_processing.jpg");
         image = ImageIO.read(input);
         width = image.getWidth();
         height = image.getHeight();

         for(int i=0; i<height; i++){

            for(int j=0; j<width; j++){

               Color c = new Color(image.getRGB(j, i));
               int red = (int)(c.getRed() * 0.299);
               int green = (int)(c.getGreen() * 0.587);
               int blue = (int)(c.getBlue() *0.114);
               Color newColor = new Color(red+green+blue,

               red+green+blue,red+green+blue);

               image.setRGB(j,i,newColor.getRGB());
            }
         }

         File ouptut = new File("grayscale.jpg");
         ImageIO.write(image, "jpg", ouptut);

      } catch (Exception e) {}
   }

   static public void main(String args[]) throws Exception
   {
      GrayScale obj = new GrayScale();
   }
}
```

Output

When you execute the given example, it converts the image **digital_image_processing.jpg** to its equivalent Grayscale image and write it on hard disk with the name **grayscale.jpg**.

Original Image

Grayscale Image