EE 356
Notes on Project 1 – ASCII Art

1. You will need to choose a non-proportional or monospaced font. Proportional fonts such as Times New Roman have differing letter widths. Courier New, Consolas, Lucida Console are three such fonts.

2. The GetPixel and SetPixel functions in the WriteableBitmap Notes on the website both have lines:

```
 if (!wbm.Format.Equals(PixelFormats.Bgra32))
    return;
```

The format Bgra32 is a color format that specifies four bytes, one for Blue, one for Green, one for Red, and one for the opacity. If you are using bitmaps generated by Paint and maybe other programs that don't use opacity the format may be Bgr32. Bgra32 and Bgr32 are identical but in Bgr32 the opacity byte is ignored. If your bitmap uses Bgr32 you will need to change the line in GetPixel and SetPixel to get them to work. If you have a writeablebitmap and you want to know what format it is in you can do this:

```
   PixelFormat pfmt = wbmap.Format;
```

3. The term *Rendering* is used in graphics to mean altering a bitmap or other graphics format to fit on an image which may be in another format. So a bitmap that is 100 x 200 pixels could be rendered to a screen that is 400 x 500 pixels. Colors and opacity levels can also be rendered. In general this does not change the original bitmap. So a bitmap that is 100 x 200 which is rendered to a screen image that if 400 x 500 remains the same size but the image on the screen will be different.

4. If you want to make ASCII art out of a color image you can convert the color image to a gray scale in at least three different methods. First you could use a program such a Paint.net (free online) to do the conversion for you. Second you could convert each pixel using the following:
R = (R + B + G) / 3
G = (R + B + G) / 3
B = (R + B + G) / 3
Third you could convert each pixel by replacing its original RGB values to the sum of 30% of the red value, 59% of the green value, and 11% of the blue value.

5. If you have a bitmap and you want to alter its pixels you will need a writeable bitmap. To convert a bitmap to a writeable bitmap you can use the following which converts your bitmap (bmp) to a BitmapSource (bitmapSource) and then converts the bitmapSource to a WriteableBitMap (wbmap).

```
  BitmapSource bitmapSource =
  System.Windows.Interop.Imaging.CreateBitmapSourceFromHBitmap
             (bmap.GetHbitmap(), IntPtr.Zero, Int32Rect.Empty,
              BitmapSizeOptions.FromEmptyOptions());
  WriteableBitmap wbmap = new WriteableBitmap(bitmapSource);
```

Alternatively, if you are reading a bmp image from a file you can read it to a bitmapimage and convert it to a WriteableBitMap like this:

```
   BitmapImage bmpimg = new BitmapImage(new Uri(fName));
   wbmap = new WriteableBitmap(bmpimg);
```

where `fName` is the file name of the bmp file including the path.

6. The graphics notes on the website (Notes on Graphics) has an example of how to create graphics on a bitmap and put it on an image. Near the end of the example there are some lines that illustrate how you can write a string a text using the formattedtext class to the graphics screen. One way to get your ASCII art onto the screen is to calculate the letters needed, assemble them into a string, and print multiple string lines to form the image using formatted text.