

## Bitmap Header Reader

### Goals:

1. Continue experience using C++ struct's and methods.
2. Continue experience reading from binary files.
3. Learn how to dynamically allocate memory.

### Tasks from Part 1:

1. Create Three data types (BitmapFileHeader, BitmapImageHeader, and Image)
  - a. The attributes of each of the header types should be the same as the file and info attributes listed at [http://en.wikipedia.org/wiki/BMP\\_file\\_format](http://en.wikipedia.org/wiki/BMP_file_format).
  - b. I will help you in class decide what to make the data type of each header attribute.
  - c. Add display methods for each of the header types. These methods should display all of the attributes of the type in an organized manner.
  - d. The Image type should have a BitmapFileHeader and a BitmapImageHeader.
2. You are to write a c++ program that will:
  - a. Create a variable of type Image named image.
  - b. Ask the user for the name of a bit map file. (i.e. junk.bmp)
  - c. Open the file for reading in “binary” mode.
  - d. Read the attributes from the file headers and store them in to the image’s header attributes.
  - e. Call the display methods on the headers.
  - f. Close the file.

### Tasks for Part 2:

1. The following items should be added to your solution to the first bitmap assignment.
2. Create a data type to represent a Pixel.
  - a. This data type should have three unsigned characters (blue, green, and red).
3. Add a pointer to a Pixel named pixels to the Image type.
  - a. This will be used to point to the list of Pixels that will be dynamically allocated.
4. After reading the file header and info header dynamically allocate width \* height Pixels using the key word new. Be sure to make the Image attribute pixels store the address that is returned.
5. Read the pixels from the file and store them one at a time into the array of pixels.
6. Write the image to a new file. Ask the user for the filename to write to.