CS-6963 - Digital Forensics
Module 7 - Programming Assignment
EXIF Parser

For this assignment, you will be writing a script in Python which parses selected EXIF fields from JPG files. The script will be submitted to Gradescope, where it will be autograded. **It is extremely important you follow the criteria exactly.**

Requirements:

- Your script must be named ‘exifParse.py’ (case sensitive)
- It must take exactly one argument, which will be a JPG file
- Your script must properly handle no argument being provided
- It must also handle an argument where the file doesn’t actually exist or is unable to be opened (hint: IOError)
- Your script must display the following criteria to stdout:
    - The name of the provided source file (JPG argument)
    - The make of the device which took the photo
    - The model of the device which took the photo
    - The date & time the photo was taken
    - The Latitude of the photo location
    - The Longitude of the photo location

You script will be graded using Python3, and the following EXIF related libraries will be available to be used:

- exifread
- exif
- piexif

When executed, your program should produce output like this:

```
root@kali:/mnt/hgfs/SHARED# python3 exifParse.py file1
Source File: file1
Make: Apple
Model: iPhone 4S
Original Date/Time: 2012:02:07 08:17:05
Latitude: 30 degrees, 5.0 minutes, 9.6 seconds
Longitude: -94 degrees, 5.0 minutes, 57.6 seconds
root@kali:/mnt/hgfs/SHARED#
```

If no argument is supplied, your program should display the message "Error! - No Image File Specified!" as follows:

```
root@kali:/mnt/hgfs/SHARED# python3 exifParse.py
Error! - No Image File Specified!
root@kali:/mnt/hgfs/SHARED#
```

If an argument is supplied, but the program is not able to open it, it should display the message "Error! - File Not Found!" as follows:

```
root@kali:/mnt/hgfs/SHARED# python3 exifParse.py file12345
Error! - File Not Found!
root@kali:/mnt/hgfs/SHARED#
```

Each different requirement will be tested and scored independently. You are being provided with two test files (file1 & file2), however the actual test will randomly select one of five files to test against. Once you submit your script, it will show you the results almost immediately, and let you know how each component was scored, as well as which test file was used. I reserve the right to rerun the autograder to verify your code properly handles any of the five files.

Lastly, unlike other assignments, you may re-submit your script as many times as you like up until the deadline, with the system grading and providing feedback each time. Only your last submission will count, so you are welcome to (and encouraged to) keep trying until you have a submission which scores 100.