



Assessed Coursework

Course Name	Cyber Security H/M			
Coursework Number	1			
Deadline	Time:	430pm GMT	Date:	22 nd February 2016
% Contribution to final course mark	20%		This should take this many hours:	20 max per person (assuming the content is understood and working in pairs)
Solo or Group ✓	Solo	If desired	Pairs	✓
Feedback Type? ✓	Written ✓		Oral	Both
Individual or Generic? ✓	Generic ✓		Individual	Both
Submission Instructions	One student should submit the .java files in a single .zip file to Moodle Name the zip file <surname1>_<surname2>.zip where <surname1> is replaced with your surname, and <surname2> is replaced with your partner's surname (assuming you have one). Penalty for non-adherence to submission instructions will apply.			
	See section 'Marking Criteria'			
Marking Criteria				
Please Note: This Coursework cannot be Re-Done				

Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below. The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

- (i) in respect of work submitted not more than five working days after the deadline
 - a. the work will be assessed in the usual way;
 - b. the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.
- (ii) work submitted more than five working days after the deadline will be awarded Grade H.
- (iii) Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause via MyCampus.

Penalty for non-adherence to Submission Instructions is 2 bands

You must complete an “Own Work” form via

<https://webapps.dcs.gla.ac.uk/ETHICS> for all coursework

UNLESS submitted via Moodle

Steganography Program

This aim of this assessed exercise is to write a steganography program in Java which achieves the following tasks:

Tasks:

1. Hiding and extracting a string from an uncompressed bitmap image file (you can assume the format of the cover image will always be .bmp) using the least significant bit algorithm.
2. Hiding and extracting any file (assuming sufficient hiding space) within a bitmap cover image using the least significant bit algorithm.

To achieve these tasks, you have been provided with two .java files on Moodle which provide a structure to be used in completion of this assessment. Steg.java has a skeleton structure, with method signatures, return types, and a description of the task each method should perform. It is the class which will perform the hiding and extraction.

FileReader.java is a class which manages reading the bytes and bits of the payload when completing Task 2. Thus, if you complete only task one you need not use the FileReader class. In FileReader more methods are filled in, with only two you need to complete – populateSizeBits and populateExtensionBits.

You must implement the methods prescribed in these files, taking the input parameters described and returning the output as described. Failure to do so will result in a minimum of 2 band reduction of your final grade. The use of the structure provided will allow us to assess your work using a program which automatically tests the success of your program by hiding and extracting files.

In addition to the methods prescribed, you can write additional methods and classes if necessary to assist in the program. However, be sure that the methods outlined take the prescribed input and provide the specified output as these will be the methods called on for marking.

This assessment can be completed as a solo project or as part of two person team. Note that if you elect to do this as a solo project, it may take more time for some aspects of the assessment.

Marking Criteria

As you progress through the different tasks, you will receive a higher mark. The more tasks you successfully implement the higher your grade will be. We will establish whether you have successfully achieved each task by two methods:

- a) We will test your code for hiding and extracting using a test program. The program will check your program can successfully hide and extract a string from an uncompressed image using different test strings and different uncompressed image files. The testing program will then perform similar checks but for hiding and extracting any file type within an image file using the LSB method.
- b) We will examine your code to determine your understanding of the steganographic methods; this includes checks such as whether you have all the component parts of the algorithm and whether each part is called in the correct order.

Achieving Task 1 will result in a maximum of a C; achieving tasks 1 and 2 will result in a maximum of an A (depending on our assessment of your understanding of the algorithm).

Your grade will be returned using a band (e.g. A3, B2, C1), not a numerical value.

If completed as part of a two person team, each person will get the same mark. If you feel your partner is not making an equal contribution, you can choose to complete the project on your own but must notify the lecturer. Note that your partner till that point can use any code written at the stage of separation.

Assumptions

When developing the program, you can assume the following:

- You can assume that all cover images will be .bmp format
- You can assume that any files to be used in the hiding or extraction process will be in the same directory as the program.
- You should assume that a number of least significant bits will be required at the start of the cover image to hide the size and extension of the file being hidden, size bits are hidden first and this should be 32 bits. Extension bits are 64 bits. Pad out with 0s on the left if needed. Assume the first 54 bytes of an image file contain data about the image and should not be altered.

Hints

Note these points when developing the program:

- All files have header information at the start. For an image this includes information such as an estimate of the size, the height and width of the image. In a windows .bmp file this is 54 bytes (see http://www.fastgraph.com/help/bmp_header_format.html for a breakdown). This header must be skipped before you get to the bytes which relate to the pixels which can be used for hiding information.
- Bytes can be treated as integers in Java
- You can cast an integer to a char in java e.g. `char(97)=a`
- Bitwise and bit shift operators in Java are discussed here: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op3.html>
- You may wish to look at ArrayLists and FileInputStream and FileOutputStream.