

OOP Sample Question 2 (RKH)

Consider a Java program that displays photos. At the core of the program is a class `Photo` that contains metadata about a given photo and the photo image data itself.

- (a) The program maintains references to the `Photo` objects within a data structure that keeps them in order of ascending file size.
- (i) Suggest an appropriate data structure class from the Java Collections framework to hold the `Photo` objects. Justify your choice. [3 marks]
 - (ii) Show how to use the `Comparable` interface to make `Photo` objects sortable by ascending file size. You may assume that the `Photo` class has a `getFileSize()` method. [5 marks]
 - (iii) The interface now supports dynamically switching the display order by selecting ordering metrics such as file size, creation date, filename, etc. Describe how you would support this within the Java Collection framework. It is not necessary to write out all the code required. [3 marks]
- (b) The `Photo` objects take up too much memory. A programmer points out that the program is either displaying a multitude of scaled-down photos or a single high resolution photo. Therefore the full image data for a particular photo can be loaded into memory on request and deleted when finished with.
- (i) Describe how to use the `Proxy` design pattern to support this idea. [6 marks]
 - (ii) Describe how memory allocated to image data that is no longer required could be reclaimed. Identify all possible outcomes of the process. [3 marks]