## 3 Further Java (arb33)

A novice Java developer designs a client-server file download service for the Internet. On start-up the server accepts a directory name on the command line and then operates in a loop, retrieving files from this directory when requested to do so by clients. Clients request the contents of a file by connecting to the server on port 1991 and sending the text string of the filename followed by a newline character. The server responds to a client request for a file by sending the contents of the file as bytes to the client before closing the connection and waiting for the request from the next client. If the client requests a file which does not exist, the server closes the connection.

(*a*) Sketch Java code for a single-threaded version of the server with blocking I/O as specified above. You do not need to handle failure due to `java.io.Exception`. You may make use of the Java standard library and you might find the following classes helpful: `java.io.BufferedReader`, `java.io.File`, `java.io.FileInputStream` and `java.io.InputStreamReader`. [8 marks]

(*b*) Describe a significant disadvantage of implementing the server as required for Part (*a*). How might this be mitigated? [3 marks]

(*c*) Another developer suggests the server is re-written to send or receive serialized Java objects in all communications between the client and the server. Briefly outline how this approach would work and provide suitable class definitions for any required serialised objects. Describe one advantage and one disadvantage of this approach compared to the method used in Part (*a*). [4 marks]

(*d*) Outline in words how you might improve this service so that clients can securely download, upload, modify and delete files stored by the server. Consideration should also be given to communicating server errors to the client. [5 marks]