7 Further Java (ARB)

Five housemates run a "status" server on their home network. The server stores the current status of each housemate as a string of text. For example, housemate Eva might set her status to "Gone to the exam hall."

Messages are passed between clients and the server as text strings sent over TCP. The new line character is used exclusively as the last character in every message. On connection with the server, a client can either (i) query the status of a user by sending the user's name to the server as a string (and the server responds with the current status message), or (ii) set the status of a user by sending the user's name followed by a colon and the new status message. For example, "Eva:Gone to the exam hall." sets the status message for Eva.

(*a*) Implement a status server in Java. The server should run indefinitely, responding to client requests. Once a client request has been fulfilled, the server should close the connection. You may assume current status messages are lost if the server is restarted and you do not need to handle exceptions. [8 marks]

(*b*) One housemate suggests the server and client should communicate by serialising Java objects rather than sending messages as text.

(*i*) Describe in words the changes you would make to your server implementation to send messages as serialised Java objects. [3 marks]

(*ii*) List two advantages and two disadvantages of an implementation based on serialised Java objects versus sending messages as text. [4 marks]

(*c*) Another housemate suggests that the server should not close the client's connection after answering the request. Instead the connection should remain open until the client sends another request or closes the connection. Describe in words what changes you would need to make to your implementation in part (*a*) to achieve this and comment on the advantages and disadvantages of this idea. [5 marks]