

## **Students should be able to...**

List and define the five layers of the Internet Model in order from top to bottom.

Explain the core responsibility of each layer.

Describe the concept of protocol layering and critique its advantages and disadvantages.

Illustrate the process of encapsulation.

Explain the "End-to-End Argument" in system design.

Differentiate between the roles of end hosts and routers, identifying which layers each device operates on.

Define what a protocol is.

~~Design a simple protocol for a new application (example: a basic chat service), defining the message format (headers and payload) and the rules for message exchange.~~

Explain the client-server model, defining the roles of the client and the server.

Compare packet switching (used by the Internet) with circuit switching (used by the telephone system).

Trace the high-level steps of a web request, from a user click to receiving data.

Distinguish between the different types of addresses used at various layers: ~~MAC addresses~~, IP addresses, and port numbers.

Define the role of HTTP in Web browsing.

Recognize the two main types of HTTP messages: request and response.

Deconstruct a basic HTTP GET request, identifying the request line, headers (including Host header) and the terminating blank line.

Deconstruct a basic HTTP response, identifying the status line (example: 200 OK), headers (example: Content-Type), and the message body containing the requested object.

Explain why HTTP is considered a stateless protocol and what that means for server design.

Define what a socket is.

Outline the sequence of socket calls on the server and the client required for a basic client-server interaction.

Understand the purpose of all socket functions (example: `connect()`), as well as inputs and outputs.

~~Develop the code for a simple client-server application that uses sockets to exchange messages.~~