(1) Why do network applications use sockets for their data transmissions (i.e., why do not network applications transmit data by themselves)?

- By leaving actual data communications to socket layer, programmers who are developing network applications can focus on their applications (thus, network applications will be developed quicker and most probably with less bugs).
- By using socket layer for data communications, data are sent and received using a standard method (i.e., sockets). This will make your network applications easier to communicate with network applications developed by other programmers.
- By separating the layer (the socket layer) that is responsible for actual data communications, it is possible and even easy to update the socket layer if there is any change in the ways data is sent or received (because even if the socket layer is updated, your network applications do not have to be changed).

(2) What is the client/server model (describe the concept of the client/server model)?

The client-server model is a model of network applications in which each network application is implemented by having two types of processes, called “server” and client”, in which a server should be always up and running, waiting for requests from clients and respond (provide responses) to clients’ requests. Clients make (send) requests to a server and receives responses from a server.

(3) What are “blocking function calls”?

Blocking functions are those functions (i.e., APIs) which stop the calling process until a blocking function completes a requested task (or fails).
(4) Why “accept” function duplicates a connection on establishing a connection?

An API, “accept”, duplicates a connection on a designated port on establishing a connection from a client because the initial port should be open to other new clients for establishing their connections.

(5) What are the two most important pieces of information “sockaddr_in structure” hold?

- Port number (TCP or UDP)
- IP address