1. How did you decide which type of socket to use? Why?

I have used the default socket type which is equivalent to,

```
sock_client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

The first argument (socket.AF_INET) in the above example specifies the type of address that sock_client can communicate with. In our case, socket.AF_INET refers to IPV4 addresses, which specifies that sock_client can only communicate with servers having IPV4 addresses.

The second argument (socket.SOCK_STREAM) in the above example specifies which communication protocol must be used to communicate. In our case socket.SOCK_STREAM specifies that communication should happen based on the Transmission Control Protocol (TCP). TCP is a connection-oriented protocol.

2. How did you choose the destination ports?

If no destination port is given as part of the URL, we choose 80 as the default port for http request because, based on RFC 1340 that's the port assigned to World Wide Web HTTP. This port will be used to send and receive unencrypted web pages by default.

3. What error handling cases did you implement?

I have handled socket.error, socket.timeout, TimeoutError and Exception

4. How does your program terminate? What happens to the TCP connection?

Program is terminated whenever the application encounters any Exception and the TCP connection is always closed since I used it along with the "**with**" block.

5. For the unsuccessful URLs, why were they unsuccessful?

The list of unsuccessful URLs are,

1. http://www.debuggerstepthrough.com/feeds/posts/default,www.debuggerstepthrough.com

   Reason: Chunked transfer encoding.

2. http://google.com:443

   Reason: Empty reply from server.

6. What happens if you try to access a site using HTTPS?

The program terminates and the error message will be printed to the terminal stating that "*HTTPS is not supported.*"