# Assignment 1

## Introduction

Most assignments for this course are practical in nature. This first assignment however is an exception to that rule, since we want to make sure you have a good grasp on the basics and have plenty of time to get used to the quirks of TCP/IP communication using the example provided during the lecture, before we dive into the more complicated stuff.

## Sufficient

Review the lecture material and answer the questions below.

1. Why are networks modelled using a layer stack?

|  |
| --- |
|  |

1. What is an IPAddress and what is the valid structure of an IPAddress?

|  |
| --- |
|  |

1. What is a loopback address and what do you use it for?

|  |
| --- |
|  |

1. What is a URL and what is its use?

|  |
| --- |
|  |

1. What is the difference between the IP protocol vs the TCP/UDP protocols?

|  |
| --- |
|  |

1. List 3 differences between the TCP & UDP protocol.

|  |
| --- |
|  |

1. What is a port and what port range should you use as an application programmer?

|  |
| --- |
|  |

1. What is the difference between a dedicated and a non dedicated server?

|  |
| --- |
|  |

## Good

Review the lecture material and answer the questions below.

1. What are the 5 layers of the discussed networking stack and what is the purpose of each?

|  |
| --- |
|  |

1. You are playing a network game where one player can shoot another player.  
   a) List the network messages (in regular English/JSON/XML) between client & server for both a client authoritative setup and a server authoritative setup for such an event from the moment a player presses the fire button.

For example if I wanted to describe a login message I could write something like  
<Login name="..." pass="..." /> or Login = { user:"...", pass:"..." } to make it clear what kind of message I want to send and what data it contains.

|  |  |
| --- | --- |
| **Client authoritative** | **Server authoritative** |
|  |  |

b) Explain whether the messages from the previous question are IP, Transport or Application protocol messages.

|  |
| --- |
|  |

1. Lecture 1 discussed two common setups for building a network game (Peer 2 peer & Client/Server). List/research some advantages/disadvantages of both setups.

|  |
| --- |
|  |

1. List which protocol type is more appropriate for each of the message types below:
   1. Login message UDP / TCP
   2. Fast paced position update messages UDP / TCP
   3. Video stream data UDP / TCP
   4. Player hit messages UDP / TCP
2. For each situation below indicate which protocol has been used (more than 1 correct answer possible):
   1. The client sends 1 message and the server receives it: UDP / TCP
   2. The client sends 1 message and the server receives it twice: UDP / TCP
   3. The client sends 2 messages and the server receives them in order: UDP / TCP
   4. The client sends 2 messages and the server receives them out of order: UDP / TCP

## Very good

Examine the ***001\_basic\_tcp\_echo\_server\_commented*** example and answer the questions below:

1. What do we mean with blocking operations?

|  |
| --- |
|  |

1. Name 2 different blocking network operations used in the given example.

|  |
| --- |
|  |

1. List some exceptions that might occur while trying to communicate over the network.  
   (Hint: check the code hinting or look up some network calls on MSDN)

|  |
| --- |
|  |

1. Imagine you've been given a client and server without any error handling code and you only have   
   time to fix one of them. Which one would you fix and why?

|  |
| --- |
|  |

1. Why does the client stop working in the given example if you send an empty string?

|  |
| --- |
|  |

## Excellent

1. Start the server and **two** clients. Why is the server only responding to the first client?

|  |
| --- |
|  |

1. Start the server and **two** clients. What is the simplest way without making any code changes to have the server respond to the last client?

|  |
| --- |
|  |

1. How can you prevent a client from connecting at all if there is already another client waiting to be served? (Hint: research the TcpListener.Start call)

|  |
| --- |
|  |

1. What happens to clients that are trying to connect, but have not yet been accepted by the TcpListener.AcceptTcpClient call?

|  |
| --- |
|  |

1. Replace line 17 of the client with:   
   TcpClient client = new TcpClient(new IPEndPoint(IPAddress.Any, 55556));  
   Start the server and two clients again and note what happens.  
   Undo line 17 and repeat, note the port the clients are connecting two.

Select the correct statement below and motivate your answer:

a) It is not possible to bind more than one TcpClient to the same port.

b) Multiple TcpClients can be bound to the same port

c) Multiple TcpClients can be bound to the same port on the server but not on the client

d) Multiple TcpClients can be bound to the same port on the client but not on the server

|  |
| --- |
|  |