

## Litteratur

J.Kurose & K.Ross, Computer Network , Pearson Education 7<sup>th</sup> ed.

<b>chap 1 Computer Networks and the Internet</b>	1.1.3 What is a Protocol + 1.2 The Network Edge + 1.3 The Network Core + 1.5 Protocol Layers and their Service models	<b>p. 35-39 + 49-59 + 75-83</b>
<b>chap 2 Application Layer</b>	2.1 Principles of Network Application + 2.2 The Web and HTTP-> 2.2.5 2.4 DNS - The Internet's Directory Service 2.5 Peer-to-peer Applications 2.7 Socket programming	<b>p. 111-136 + 154-163 + 168-175 + 185-198</b>
<b>chap 3 Transport Layer</b>	3.1 Introduction and Transport-Layer Services 3.2 Multiplexing and Demultiplexing 3.3 Connectionless Transport UDP 3.5 Connection-Oriented Transport: TCP	<b>p. 215-234 + 261-269 + 272-289 + 297-300</b>
<b>chap 4 Network Layer</b>	4.1 Introduction 4.3 The Internet Protocol (IP)	<b>p. 333-336 + 357-360 + 362-379</b>
<b>chap 6 DataLink Layer</b>	6.1 Link Layer Introduction and services	<b>p. 467-472</b>
<b>chap 8 Security in Computer Networks</b>	8.1 What is Network Security 8.2 principles of Cryptography 8.3 Message Integrity 8.6 Securing TCP Connections: SSL 8.9 Operational Security	<b>p. 621-628 + 632-634 + 637-649 + 659-663</b>

## Serialization

JSON

- <https://www.json.org/>

## web services - REST

REST

- [Note om REST](#)
- [RESTful Service Best Practices- URL design](#)

Cross Origin  
Reference

- [Note om Cors](#)

## Tools ([WEB-Tools](#))

- Wireshark - Packet sniffer
- Fiddler - Packet sniffer
- Postman - HTTP-client
- SocketTest

Azure account (<https://portal.azure.com/>)

## Opgaver

- TCP-server | TCP Client
  - Simpel TCP program <http://pele-easj.dk/2020e-tek/exercises/socket/EchoServer.htm>
  - Concurrent TCP-server <http://pele-easj.dk/2020e-tek/exercises/socket/EchoServerConcurrent.htm>
  - Mere TCP programmering <http://pele-easj.dk/2020e-tek/exercises/socket/MoreTCPprogramming.htm>
  - TCP programmering – JSON <http://pele-easj.dk/2020e-tek/exercises/socket/SocketXMLandJson.htm> (kun json del)
- UDP sender | UDP modtager
  - UDP programmering <http://pele-easj.dk/2020e-tek/exercises/UDPServerClient.htm>
  - UDP Proxy <http://pele-easj.dk/2020e-tek/exercises/UDP-Proxy.htm>
- SSL server | SSL Client
  - Certifikat #1 <http://pele-easj.dk/2020e-tek/exercises/ssl/CertificateX509.pdf>
  - Certifikat #2 <http://pele-easj.dk/2020e-tek/exercises/ssl/CertificateX509%20No.2.pdf>
  - SSL-programmering <http://pele-easj.dk/2020e-tek/exercises/ssl/SSL-Secure%20Socket.pdf>
- REST-Service (opgaverne 1-7)
  - Simpel REST service <http://pele-easj.dk/2020e-tek/exercises/RETSservice1-simple-CRUD.pdf>
  - Ny route til ny metode  
Query <http://pele-easj.dk/2020e-tek/exercises/RETSservice2-advanced.pdf>
  - Help pages – swagger <http://pele-easj.dk/2020e-tek/exercises/RETSservice3-swagger.pdf>
  - Testing unittest hhv. postman <http://pele-easj.dk/2020e-tek/exercises/RETSservice4-Testing-and-Azure.pdf>
  - C# console app consuming REST <http://pele-easj.dk/2020e-tek/exercises/RETSservice5-ConsumingRest.pdf>
  - CORS <http://pele-easj.dk/2020e-tek/exercises/RETSservice6-Cors.pdf>
  - Database <http://pele-easj.dk/2020e-tek/exercises/RETSservice7-Database.pdf>

## Råd og vejledning til eksamen

### Før eksamen:

- Læs litteratur (benyt begreber1 + begreber2 for at tjekke om du kan stoffet)
- Regn opgaverne – og noter dig, hvor du har løsningerne.
- Regn de tre prøveopgaver (Mock1, Mock2 og Mock3)

### Optil eksamensdagen:

- I Azure hav 2 app-services og én database klar
  - App service 1 – til rest
  - App service 2 – til Typescript
  - Database (kun én skal blot lav én tabel)
- Start visual studio (lad vær med at opdater Visualstudio lige op til eksamen – eller under)
  - Hav evt. et REST-projekt klar
- Download og gør klar til typescript
  - Hav et typescript klar dvs. clone fra git + npm install