

## Introduktion till projekt

TCP: stream-baserat    UDP: datagram-baserat  
(ström)

Mottagaren måste tolka de byte som tas emot.

InputStream/OutputStream används i java för att ta emot/sända byte  
- argument kan vara buffer (med/utan offset & längd) eller int (bara för output)  
- man måste alltså själv skapa buffrar

Konvertering måste göras mellan datatyper och byte    Encoding/Decoding  
(Kodning/Avkodning)  
- Flera kodningsstandarder  
- ASCII - en byte per tecken  
- UTF-8 (Unicode Transformation Format) - Variabel längd på tecken (1-4)  
- EBCDIC, ISO 8859, MS-Windows, Mac OS Roman, UTF-16...

String.getBytes och new String kan användas.

- kodningsstandard ges som argument, t.ex. StandardCharsets.UTF\_8

Det finns många olika standarder även för andra datatyper  
Såsom heltal, flyttal, m.m.

## Socket-programmerings-projekt

- Socket-programmering i Java
- TCP-klient och -server
- HTTP
- Uppdelat i flera inlämningar (en veckas mellanrum)
- Automatisk rättning - följ instruktioner, mallar finns.
- Handledningstillfällen - boka

1. TCPAsk - klient som kontaktar en server och skriver ut svar.
2. HTTPEcho Server - server som skickar tillbaka data inbakat i ett HTTP-svar (så att HTTP-frågan visas)
3. HTTPAsk Server - en HTTP-server som implementerar TCPAsk och returnerar svaret som ett HTTP-svar.
4. Multi-threaded HTTPAsk server. Som 3. fast med möjlighet att hantera flera klienter.

Samarbeta gärna men lämna in individuellt. (Kopiera inte)

Eget ansvar att testa och debugga