**ITGK – øving**

**Starting out with Python**

Kap.1 Introduction to Computers and Programming

1.2 hardware & software  
Hardware – Central prossesing unit (CPU), hovedminne, sekundærminne, input og output enheter  
CPU – kjører programmer. I dag er de små chiper kjent som mikroprosessorer,  
RAM - random acsess memory, volatile, minne forsvinner når du skrur av  
Sekundærminnet – platelager(harddisk HD eller hard disk drive HDD) spinnende disk sekvensielt  
Sekundærminnet – SSD solid state drive, dyrere, mer stabilt, ingen bevegelse, «flash memory»  
ekstra lagring USB «universial serial bus» (minnepen), CD compact disc, DVD (digital versatile disc)  
input devices – datamus, keyboard, touchskjerm, mikrofon, digitalkamera etc.  
output devices – skjermer og printere  
Software – system software eller applikasjons software  
system software – Operating systems ((OS) Android, iOS,windows, unix, macOS, Linux), utility programs ((systemverktøy) virusskannere, filkompresjons program, backup program) og Software development tools (programmeringsverktøy)  
Applikasjons software - Gjør PCen brukbar for hverdagsutfordringer, email, Word, spill etc.

1.3 Hvordan datamaskiner lagrer data  
- All lagret data er lagret binært.  
- Lagres i bytes som er 8 bits  
- ASCII syv byts – American standard code for information interchange  
- Unicode ny standard som tar med bokstaver fra andre språk  
- negative tall toer kompliment  
- flyttall lagres i floating-point notasjon  
- digital data er data lagret binært

1.4 Hvordan et program virker  
- CPUen gjør følgende leser data fra hovedminnet, adderer to tall, trekker et tall fra et annet, multipliserer to tall, deler et tall på et annet, flytter lagret data fra et sted til et annet, avgjør om noe er likt  
**Fetch-execute cycle**

1. Instruction **Fetch** (IF)
2. Instruction **Decode** (ID)
3. Data **Fetch** (DF)
4. Instruction **Execution** (EX)
5. Result Return (RR)

- Assembly enkleste form for programmeringsverktøy som ikke er 0ere og 1ere. Bruker en assembler for å gjøre ord til 0 og 1. assembly er low-level language  
- high-level language: Python (1990s) ADA, BASIC, FORTRAN, COBOL, PASCAL, C, C++, C#, Java, JavaScript, Ruby, Visual Basic

Python keywords:  
- and, as, assert, break, class, continue, def, del, elif, else, except, False, finally, for, from , global, if import, in is, lambda, None, nonlocal, not, or, pass, raise, return, True, try, while, with, yield  
  
Program development cycle:  
designe programmet 🡪 skriv koden 🡪 rette opp syntax feil 🡪 Teste programmet 🡪 rette opp logiske feil 🡪 designe programmet 🡪 etc.

Algoritme – et sett av veldefinerte logiske steg som må gjøres for å utføre en oppgave  
Pseudocode – falskkode

Enkel Kode:  
print(‘’hei’’) eller print(‘hei’)  
#for å skrive kommentarer  
var = verdien til variablen  
int(heltall), float(kommatall),   
var=input(prompt)

Mattesymboler:  
+ addisjon  
- subtraksjon  
\* multiplikasjon  
/ divisjon (gir resultatet som et flyttall)  
// heltallsdivisjon (heltallsdivisjon runder ned!) (ved negative tall bort fra null) 5//2=2 eller -5//2=-3   
% modulo eller rest (gir resten)  
\*\* eksponent

Print(kode):  
print(‘hei’, end=’ ‘)  
print(‘på deg’, end=’ ‘) 🡪 hei på deg

Print(‘one’, ’two’ , ’three’, sep=’\*’) 🡪 one\*two\*three  
Print(‘one’, ’two’ , ’three’, sep=’~~~’) 🡪 one~~~two~~~three  
print(\n enter, \t tab, \’, \’’, \\ ,gir tegnene bak \)  
format(12345.6789, ‘.2f’)

Kap 2. turtle (uinteressant)

Kap 3. valg struktur og boolean logikk  
Boolean - True or False  
matteutrykk - >,<,>=,<=,==,!=

If 🡪 elif 🡪 else  
logiske operatorer: and, or, not

Kap 4. repetisjons strukturer  
While for element in range or for element in liste etc.  
sentinel verdi som markerer slutten (f.eks liste med høyder kan man bruke 0 som sentinel)  
nested loop (flere for løkker inni hverandre)

Kap 5. funksjoner  
enklere kode, gjenbruk av kode, bedre testing, raskere utvikling, enklere med samarbeid  
top-down design bryter ned utfordringene til mindre utfordringer  
Hierarki eller strukturgraf forklarer hvordan funksjonene påvirker hverandre  
Globale konstanter med store bokstaver:

From random import 🡪 randint, randrange(5,10,2)(fra 5, men ikke med 10, annenhvert tall), seed(10) (får de samme tilfeldige tallene hver gang etter vi kjører seed() med den verdien)

IPO-chart 🡪 input, processing, output (sier noe om hva en funksjon gjør)

From math import 🡪 cos(x),acos(x),ceil(x),floor(x),sqrt(x),log(x),log10(x),exp(x),math.pi,math.e etc.

Kap 6. Files and exceptions  
Open file 🡪 process the file 🡪 close the file  
file\_var=open(filename, mode) modes: ‘r’,’w’,’a’ etc. 🡪 file\_var.close()  
try 🡪 exception 🡪finally (kjører selv om en exception har kjørt)

Kap 7. Lists and tuples

En liste kan inneholde forskjellige typer (float,int string etc.)  
Husk å ikke endre lister, men lag en ny liste med den verdien hvis du skal bruke de i original form senere.   
slice, liste[start:end:jump] tar ikke med verdien i end  
viktige liste funksjoner:  
liste.append(item) (legger til på slutten av listen)  
liste.index(item) returnerer index til første element med verdi item. ValueError hvis ikke funnet.  
liste.insert(index, item) setter inn item i index plass  
liste.sort() sorterer i stigende rekkefølge  
liste.remove(item) fjerner første tilfelle av item i listen  
liste.reverse() endrer rekkefølgen

del list[2] (fjerner verdien i den posisjonen)  
min(liste) eller max(liste) finner min eller max verdi

kap 8. mer om strenger  
Man kan itrere gjennom en streng: for ch in name  
strenger ikke muterbare  
kan bruke len(streng)  
kan slice strenger, men da må de gis en ny variabel first\_name=full\_name[:5]  
noen funksjoner:  
isalnum() True hvis bare bokstaver eller tall og er minst en i lengde ellers False  
isalpha() True hvis inneholder bare bokstaver og er minst en i lengde eller False  
isdigit() True hvis tall eller false  
islower() True hvis alle er små bokstaver  
isspace() True hvis den inneholder kun space, \n,\t og er minst en i lengde  
isupper() True hvis den innholder kun store bokstaver og er en i lengde eller False  
string.lower(), string.upper(), string.strip(), string.strip(char)  
string.replace(old, new),

kap 9 dictionaries og set  
telefonbok={}

if ‘Chris’ in telefonbok:  
print(telefonbok[‘Chris’])  
telefonbok[‘kåre’]=9415578 legger til key og value  
del dictionary\_name[key]  
value=dict\_name.get(‘Katie’, ‘Entry not found’)  
myset=set()

myset.add(1)  
print(myset)🡪 {1}

myset.update([3,4,5])  
print(myset)🡪{1,3,4,5}  
noen funksjoner:  
set1.union(set2), set1.intersection(set2), set1.difference(set2), set1.symmetric\_difference(set2), set1.issuperset(set2), set1.issubset(set2),

serialisere objekter  
🡪 bruker pickle  
import pickle  
  
Kap. 10 classe og objektorienter programmering (ikke relevant)  
Kap. 11 inheritance (ikke relevant)

kap. 12 rekursjon  
En rekursiv funksjon kaller seg selv.  
looper er oftest bedre å bruke en rekursjon, men i noen tilfeller som fakultet eller gcd() kan det være bedre å bruke rekursjon.

Import:

From math import sqrt() 🡪 bruker sqrt() som en vanlig funksjon.  
from math import \* 🡪 dumt siden du henter alle funksjoner du ikke trenger fra math pluss at noen av de kan være funksjoner du ikke vet hva er.  
import math 🡪 bruker math.sqrt() f.eks.  
import math as mt 🡪 bruker mt.sqrt() (greit hvis bibliotekets navn er langt)  
from math import sqrt as square\_root() 🡪 bruker da square\_root()

Svar til sjekk som er relevant:  
- Et program er et sett med instruksjoner som en pc følger for å gjøre en oppgave.  
- Hardware er alle fysiske deler eller komponenter som en pc er satt sammen av.  
- En Pc består av fem hoveddeler: CPU, hovedminne, sekundærminne, input og output enhet  
Gadd ikke flere

**Theory book IT intro**

**Kap.1 definere informasjonsteknologi**

8 viktigste utviklingene i digital beregning:  
- Digitalisering av informasjon: Herman Hollerith, hullkort (punch card), manntallslister, ikke datamaskin  
- Lagret- program dataer: lagre program i software ikke hardware, CPU, lettere å endre program, program kan bli mer kompleks, autonom kan kjøre av seg selv  
- Transistorer: transistorer, mindre strøm, mindre varme, veldig pålitilig, små i vekt og størrelse, billigere, mer kompleks CPU  
- Integrated Circuits: IC integrerte kretser, «chips»-laget av silisium, doping av silisium, fotolitografi, «printe» kretsene på chippen, printing er like billig selv om det er komplekst system,   
- «Personal computers»: Moore’s law, få som så nødvendigheten av en pc i starten, barna var ofte mer lærevillig og det var starten på teknologigapet vi ser mellom gamle og unge  
- The Internet: Først ARPAN, i starten «dial-up-connection»,   
- HTTP og the world wide web: felles http protokol, CERN, Berners-Lee  
- Layered software development: før var det mye spagettikode, lettere å lage kode som bygger på gammel kode,

Ved teknisk assistanse (tech support) er det viktig å bruke riktig begreper.

Moores lov: Den sier at antall transistorer på et areal dobles hver 24. måned.

Mer teknisk navn for en datamaskin er en prosessor.

En prosessor består av CPU, litt minne kjent som cache, input/output enhet.

ARM (Advanced RISC Machine) er en prosessor som er spesialisert til å være del av et annet system. RISC (Reduced Instruction-Set Computer) Disse ARMene brukes slik at man kan kode noe til å gjøre en oppgave. Dette er fordi det er lettere med software enn hardware.

Software: samlebegrep for programmer. Program er en samling datamaskin instruksjoner som implementerer en funksjon eller handling.   
Programmere: instruere en agent til å gjennomføre en funksjon eller handling.  
Agent: noe som kan følge en instruksjon.  
Algoritme: er en presis og systematisk metode for å produsere et spesifisert resultat.   
5 egenskaper til en algoritme: input spesifisert, output spesifisert, bestemt, effektiv, endelig

Boot: starte en datamaskin. reboot: restarte omstarte en datamaskin.   
ideord: abstrakt, generalisere, operativt tilpasset, mnemonic

Mennesker:

Hollerith(1860-1929) – Zuse([1910](https://no.wikipedia.org/wiki/1910)-[1995](https://no.wikipedia.org/wiki/1995) ) – Von Neumann([1903](https://no.wikipedia.org/wiki/1903)-1957) – Engelbart([1925](https://no.wikipedia.org/wiki/1925)-[2013](https://no.wikipedia.org/wiki/2013))

Herman Hollerith - Hullkortmaskin   
Konrad zuse - Hans største bragd var konstruksjonen av den første funksjonelle [datamaskin](https://no.wikipedia.org/wiki/Datamaskin) med programmer lagret på tape, kalt [Z3](https://no.wikipedia.org/wiki/Z3), i [1941](https://no.wikipedia.org/wiki/1941).

John von Neumann – spillteori, numerisk analyse, design av de første programmerbare datamaskinene. Lagde IKKE John Von Neuman arkitektur  
Douglas Engelbart - Her deltok han i utviklingen av det sivile internettet, og på 1970-tallet patenterte han PC-musen.

Andre Mennesker:   
Alan Turning (1912-1954), har gitt vesentlige bidrag innen informatikk og la grunnlaget for en formalisering av konseptet algoritme og databehandling med begrepet turingmaskin, som er en modell for en generell datamaskin.  
Gordon Moore (1929-): Moores law  
Jack Kilby: Oppfinneren av integrerte kretser

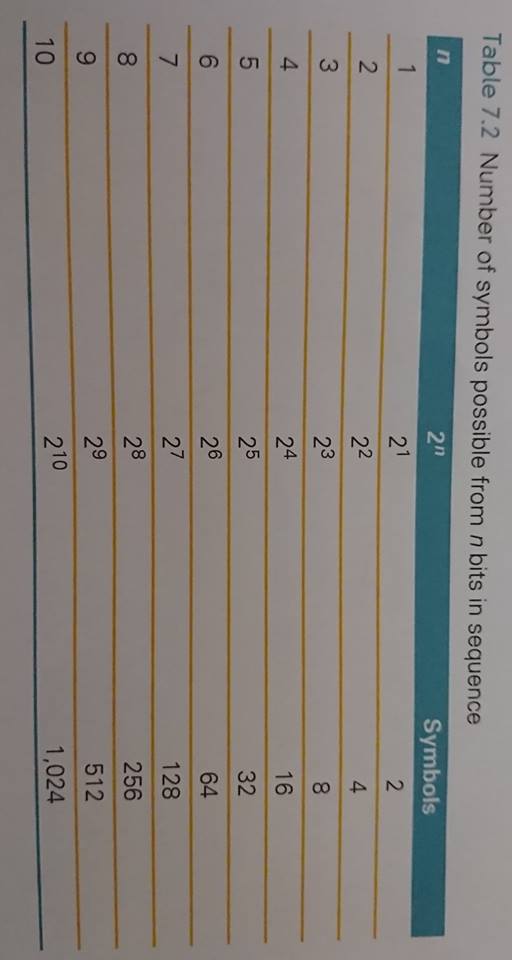
**Kap. 9 prinsippene om datamaskin operasjoner**

Prosessor=data=mikroprosessor=sentral prosessor (CP) = sentral proses enhet (CPU)=kjerne (core)  
Operativsystem (OS), software, instruksjoner, hente/utføre-syklusen, minne, hardwarekomponenter   
software stack=kode-«grunnmuren» som man kan bruke for å skrive høyspråkkode  
Shared effort = bruke andres kode til å løse problemer f.eks. funksjoner i import math

The fetch/execute cycle (hente/uføre-syklusen)  
1. Instruksjon Fetch (IF) hente oppgave  
2. Instruction Decode (ID) dekode oppgave  
3. Data Fetch (DF) hente data  
4. Instruction Execute (IE) gjøre oppgaven  
5 Result Return (RR) returnere resultatet

Alle pcer har fem basis deler: minnet, kontroll enhet, ALU, input enhet, output enhet  
  
minnet bygget opp av 1 byte 🡪 minnet har adresse 🡪 minnet lagrer en verdi 🡪 har en endelig størrelse  
1 byte nok til å holde et ASCII-symbol, eller et heltall mindre enn 256  
4 bytes memory words veldig vanlig lengde på en adresse.  
kilo-mega-giga-tera alle kilobyte ikke 1000, men 1024.   
  
RAM random accsess memory. Random fordi den kan referere til minnelokasjon i hvilken som helst rekkefølge. Angis i MB eller GB.   
CU (control unit) hardware biten av fetch/execute-cycle.   
instruksjon register: Hvor CU holder instruksjonene den jobber med  
program teller (PC): Hvor CU finner minne til neste instruksjon som CUvil jobbe med  
ALU(Arithmetic/logic unit) gjør matten i IE steget av F/E-syklen.   
input og output enheter: monitor,tastatur,mus, men også CDer og minnepenner  
Device driver: alle enhetene har en driver som gjør at PCen skjønner hva en input/output enhet gjør.  
ADD 4000, 2000, 2080 (adderer verdi i minne lokasjon 2000 og verdi i minne lokasjon 2080 og lagrer det i minnelokasjon 4000)

PC kan være forkortelse for Program counter, printed circuitboard eller personal computer…   
Branch and jump 🡪 Hvordan program counter finner ny intruction.   
Klokken: Et ‘’tick’’ er et steg i F/E-syklen. Måles i gigahertz.  
Pipeline: For hvert ‘’tick’’ vil en ny instruksjon i F/E-syklen starte. Flere oppgaver kjører samtidig (dette er pipeline).   
Translation: source code 🡪 assembly code 🡪 binary code  
source code: python eller lignende  
assembly code: veldig «enkelt» språk ikke mye ekstra funksjoner, mye må skrives som 0 og 1.  
binary code: 0 og 1 som maskinen skjønner.  
Compiling: er det å gjøre om f.eks C# kode til assembly kode.  
Assembling: assembler gjør assembly kode om til binærkode  
Integrerte kretser (IC): på grunn av photolitografi, små, billig og sikre. Forminskning og integrering er god stemning når det kommer IC.

**Kap 7. Representere informasjon digitalt**PandA – presence or absence – True/False, ja/nei, av/på, +/-, 1/0, black/white, for/against etc. de er eller det er ikke.   
En bit er informasjonen gitt i en PandA observasjon.  
Addere binært og hexadesimalt bruk «barneskolen»-måten å addere.  
Heksadesimaletall: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F  
ASCII: 7-bit American standard code for information interchange  
ISO-8859-1: 8-bit/1 byte for å få med andre tegn en de amerikanske, laget av IBM og navnet byte kommer derfra  
Unicode: et tegnsett som har som formål å skape et standard tegnsett for alle datamaskiner som støtter alle språk som er i praktisk bruk. UTF-8 e.l.  
4 bits = en nibble  
grunnen til at det heter byte er fordi det er feilkorrigering av bit. Bit🡪«bite»🡪byte (to bokstaver annerledes og du kan ikke skrive feil meollom bit og byte!

**Kap. 8 representere multimedia digitalt**RBG – red,blue,green – er for lys som «skinner ut fra» en pc eller lommelykt etc.  
ADC – analog to digital converter f.eks mikrofon  
DAC – digital to analog converter f.eks høyttaler  
Nyquist regel – dobbel samplingrate av det som spilles inn.  
mennesker hører ca. 20 000Hz 🡪 40 000Hz opptak, men vi bruker 44 100Hz fordi data.   
Flere bit gir bedre digitalisering av lyd. Standard er 16bit   
1 minutt lydopptak 🡪 60sek\*44100Hz\*16bit\*2stereo=84672000bits=10584000bytes  
MP3 komprimerer lyd slik at .wav-filen bare tar med det vi hører. Det gjør at de kun tar ca. 1/10 plass.  
Hvor stort er et bilde med disse kravene 8x10 bilde, 300pixel per inch, 3 byte per pixel = 8\*10\*300^2\*3=21.6MB  
kompresjon= lossy vs. lossless 🡪 lossy kan ikke gjøre tilbake til orginalfil f.eks MP3 eller JPEG eller MPEG  
Captcha eller reCaptcha bruker utydelig skrift for å skille mennesker fra bots  
Multimedia utfordringer: forsinkelse, båndbredde,   
en bitstreng kan være Tall, Farge, Maskin instruksjoner, ASCII symboler, lyd, IP-adresse, hextall

**Kap. 5 algoritmer**Algoritme en mengde sted som definerer hvordan en oppgave utføres.  
An algorithm is an ordered set of unambiguous, executable steps that defines a terminating process.  
En algoritme er en ordned mengde av entydig utførende steg som definerer en endelig prosess.  
Dataforskere bruker ordet effektiv for noe som er «gjørbart» effecitv 🡪 doable  
Pseudocode mindre formell kode som brukes til å forklare hvordan en kode oppfører seg.  
For å vise kode brukte man ofte før flowcharts. Nå representeres det oftest med tekst.  
navngiving av variabler: camelcase (areYouCool), pascalcase (AreYouCool), underline(are\_you\_cool)  
top-down methodology (generelt til spesifikt) bottum-up methodology (spesifikt til generelt)  
insertion sort: worst O(n^2)  
bobbel sort: worst O(n^2)  
Selection sort: worst O(n^2)  
bianary search algorithm: O(logn)  
sekvensielt søk: O(n)  
iterative funksjoner leter først etter terminerende hendelse (base case eller degenerative case):

**Kap. 1 Nettverk**5 hovedaspektene av nettverk:  
- nettverksapplikasjon og nettverksprogrammering  
- Datakommunikasjon  
- pakkeswitching og netverksteknologier  
- internettarbeid med TCP/IP  
- tilleggs nettverikingskonsept og teknologier  
Pakkeswitching fremfor linjeswitching  
ISP - internett service provider  
private network – consumer, small Office Home office (SOHO), Small-to-medium Business (SMB), Large enterprise.  
Kommunikasjonsstandarder setter gjennom protokoller lavt-nivå f.eks volt høyt-nivå formatet til meldinger.  
TCP Transmission control protcol (Internet Protocol Suit)  
IP Internet Protocols  
TCP/IP stack:  
lag 5: Application – hvordan tjenester samhandler under kommunikasjon(HTTPS,SMTP,WWW,snapchat)  
lag 4: transport – maks hastighet motta data, passe på at data mottas i riktig rekkefølge (TCP,DCCP,SCTP, ntnu.no,google.no)  
lag 3: Internet – spesifiserer kommunikasjon mellom to datamaskiner (IP,IPv4,IPv6,ICMP)  
lag 2: Network interface – samhandlingen mellom nettverks hardware og lag 3 (MAC)  
lag 1: Phsycal – hardware, elektriske egenskaper, radio frekvenser   
IPv4 vs. IPv6 begynt å bruke v6 fordi flere IP-adresser.

**Kap. 2 Internett trender**skifte i hva vi sender fra tekst til høykvalitets multimedia  
tekst 🡪 grafiske bilder 🡪 videoklipp 🡪 HD video  
Notifikasjonslyder 🡪 menneskestemmer 🡪 lydfiler 🡪 HD musikk  
telefonsystemet: fra analog til VoIP (voice over IP)  
tv-systemet: fra analog sending til IP sending  
mobilnett: fra analog til 4G  
internett kobling: ledninger til trådløs  
data aksess: sentralisert til peer-to-peer P2P eller likemansnett  
sosialt nettverk, sensornettverk, videokonferanser, nettbank og betaling  
skytjenester: skytjenester ofte elastiske betal for det du bruker

**Kap. 8 pålitelighet og kanalkoding**

3 hovedgrunner til sendingsfeil:  
- interferens: radiostøy fra andre enheter eller universet  
- forvrenging: endring på det som sendes. F.eks metaller eller magneter nærheten av ledninger  
- demping/svekking: når et signal passerer gjennom et medium blir det svakere over lange distanser.

Shannon’s Teori: øke signal-til-støy forholdet (enten øke signal eller minske støy)

|  |  |
| --- | --- |
| Feiltyper: | forklaring |
| Singel bit error | En bit er endret (ofte pga. veldig kort interferens) |
| Burst error | Flere bits er endret (ofte pga. lengre interferens) |
| Erasure(ambiguity) norsk tvetydighet | Usikker på om det er enn 1 eller 0 |

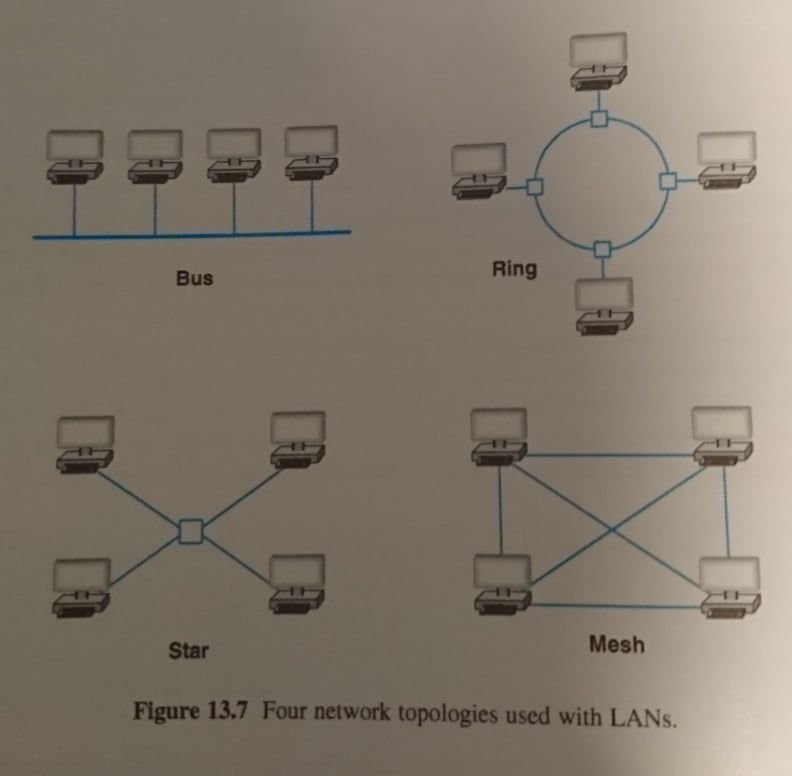
Burst size/length lengden på feilen bitsene som blir ødelagt.   
channel coding: teknikker for å overkomme datafeil (FEC eller ARQ)  
basic error correction: sjekker om en feil har skjedd  
FEC (forward error correction): sjekker om feil og endrer feil til riktig(BEC,CEC)  
ARQ (Automatic repeat reQuest): sender og mottaker sender fram og tilbake slik at det blir rett

Block Error Codes: legger på overflødighet (redundacy)for å sjekke feil, men er memoryless da den ikke tar med den overflødigheten videre.  
Conventional Error Codes: legger til memory, men har en større mulighet til å oppdage feil.  
Singal Parity Check: form for Block Error Code, partalls- eller oddetallsparitet. Oppdager ikke bursterror  
rekke og rad paritet (RAC): Row and column paritycheck kan rette opp enkel bits feil, ved bruk av «matriser».  
16-bit checksum eller internett checksum: deler en melding in i 16-bit bolker, regner ut den aritmetiske inverse av 1’er-komplimentene summen av blokkene. Heltalls overflyten (overflow) adderes tilbake i checksumen.

Cyclic Redundancy codes (CRC): high-speed datanettverk, tre kvaliteter: kan da en vilkårlig lengde, utmerket feiloppdaging, raskt å regne ut for hardware  
Hamming-avstand et mål på forskjellen mellom to strenger. Begrepet ble introdusert av Richard Hamming i 1950, og kan defineres som det minste antall utskiftninger som er nødvendig for å transformere den ene strengen til den andre. For eksempel har de binære strengene "01" og "10" en Hamming-avstand på to, da to biter må byttes om for å komme fra den ene til den andre.  
  
**Kap. 13 Local Area Networks: packets, frames og topologier**

LAN – local area network  
MAN – Metroplitan Area network  
WAN – wide area network  
linjesvitsjing vs. pakkesvitsjing  
linjesvitsjing: i dag brukes viral punkt-til-punkt kommunikasjon, egne steg for opprette, bruke og slutte.  
pakkesvitsjing: basis for internett. Meldingen deles i små pakker som sendes over nettet, tilfeldig asynkron kommunikasjon, ikke noe oppsett behov før kommunikasjon, ytelse varierer pga statistisk multipleksing.  
Innenfor telekommunikasjon, er multipleksing (MUX-ing) det å kombinere to eller flere informasjonskanaler i en felles transmisjonsmedium ved å bruke maskinvare kalt en multiplekser (eller MUX).

LAN topologier(eller form):



Bus – bare en pc kan sende av gangen, krever mindre ledninger enn en stjerne, men samme problem som en ring  
ring – hvis en pc kobler av kobles alt ned med mindre man bruker en annen enhet til å være ring. Lettere å koordinere adgang og å oppdage om nettverke oppfører seg riktig.  
Mesh – dyrt må ha for n PCer n!/((n-2)!\*2!)=(n^2-n)/2 kablinger  
Star – senter er en hub, er ikke et stort problem hvis en ledning blir skadet.   
Ved LAN brukes oftes bus,ring og star.  
Media Access Controll address (MAC address)

IEEE (Institute of Electrical and Electronics Engineers) adresseskjema støtter tre typer av adresser som tilsvarer tre typer av pakke sending:

|  |  |
| --- | --- |
| Adreese type | Mening og pakkelevering |
| Unicast | Gjennkjenner en ensling pc, og spesifiserer at kun den pcen skal motta en kopi av pakken |
| Broadcast | Korresponderer til alle pcer og spesifiserer at alle på nettverket skal ha en kopi av pakken |
| Multicast | Identfiserer et subset av pcer på et gitt nettverk, og spesifiserer at hver pc i subsettet skal motta en kopi av pakken. |

**Kap. 20 internetworking: concepts, architecture og protocols**

Ingen nettverksteknologi er best for alle bruk. LAN,MAN eller WAN.  
Et kommunikasjonssystem som bruker universal tilgang (universial service) tillater tilfeldige par av pcer å kommunisere.  
internetwork eller internett er et nettverk av nettverk.   
For å koble nettverk bruker man en ruter (eng. router).   
Ruter kan koble to LAN, LAN og WAN eller to WAN. Ruter kan koble ethernet til Wi-Fi nettverk.  
Bedrifter bruker sjelden en ruter for å koble til et nettverk fordi man kjøper gjerne flere mindre billige og bruke flere skaper stabilitet hvis en ryker.   
Intranet er internett for et selskap. Ofte ligger det på nettet slik at det i dag er vanskeligere å skille.

Host eller vert er slutt systemet som kobler til Internett og kjører applikasjoner.   
Både rutere og verter trenger TCP/IP protokoll software, men ruter trenger ikke lag 5.

**Kap. 21 IP: Internet adressering**IPv4 vs IPv6. Ikke nok plass på IPv4 som brukte 32 bit.   
Vanskelig bytte siden IP er viktig for all internett kommunikasjon, endre IP krever at man må endre hele internett.   
Applikasjoner bruker IP addresser for å kommunisere.   
IPv6 bruker 128bit adresser

IP adresser skrives på formen 129 . 52 . 6 . 0 der tallene kan være opp til 255.   
Får å løse problemet med IPv4 brukte man først subnet addressering og klasseløs addressering  
CIDR classless inter-domain routing.  
IPv6 addresser skrives i colon hex f.eks 69DC:8864:FFFF:FFFF:1280:8C0A:FFFF siden veldig mange strenger vil være 0000 så er det vanlig å fjerne de og bare skrive :: slik at   
69DC:0:0:0:0:0:FFFF blir 69DC::FFFF

**Kap. 25 TCP: Reliable Transport Service**i Internett er TCP en transport-lags protokoll som sikrer pålitelighet.  
TCP 7 egenskaper:  
- forbindelsesorientering, punkt-til-punkt kommunikasjon, fullstendig pålitelighet, full dupleks (sende begge veier samtidig) kommunikasjon, strøm grensesnitt, pålitelig forbindelses start,

grasiøs tilkobling avstengning

Hovedproblemene med sending er:upålitelig kommunikasjon, mottaker reboot, forskjellig kraft i sender og mottaker (utsender sender så raskt at mottaker ikke greier å ta imot), forstoppelse i Internett

Sequencing – hindre upålitelig kommunikasjon ved å gi hver pakke et nummer.  
positiv acknowledgement with retransmission – sjekker for pakketap med acknowledgement (ACK). Når en sender, sender en pakke starter han en klokke og venter på ACK. Hvis ingen ACK ila tid sender han en ny pakke.   
Sliding window – sender flere pakker omgangen og mottar flere ACK for å spare tid.

**Kap. 27 nettverks ytelse (QoS and DiffServ)**

PAN,LAN,MAN,WAN  
Hva måler man nettverksytelse i?:

|  |  |
| --- | --- |
| Mål | Forklaring |
| Latency(delay) – forsinkelse | Tiden det krever å sende data over et nettverk |
| Troughput(capacity) – gjennomstrømming | Mengden data som kan sendes per tid |
| Jitter(variability) – forandring/varianse i pingtid | Endringen i forsinkelse og lengden på endring |

Bredbånd brukes av dataingeniører for analog båndbredde, mens gjennomstrømming brukes for digital  
QoS – quality of service –

**Kap. 29 nettverks sikkerhet**

|  |  |
| --- | --- |
| Problem | Forklaring |
| Phishing | Late som om man er en kjent side for å få info. |
| Feilaktig fremstilling (misrepresentation) | Falske eller overdrevne utsagn om goder eller tjenester, eller gi falske eller dårligere produkt. |
| Scams – svindel | Lure naive brukere til å bruke penger eller medvirke i kriminalitet |
| Denial of service (Dos) – tjenestenekt | Blokke en side for å hindre business |
| Loss of controll – tap av kontroll | En bruker tar over PCen for å begå lovbrudd |
| Loss of data – tap av data | Miste data eller info |

Handlinger brukt i sikkerhetsangrep

|  |  |
| --- | --- |
| Teknikk | Forklaring |
| Wiretapping eller snooping – avlytting | Lage kopier av pakker for å få info |
| Replay - gjennbruk | Sende tidligere pakker som f.eks passord for å få tilgang |
| Buffer Overflow – mest vanlig | Sende mer data en mottaker forventer |
| Address spoofing - | Falske en IP addresse slik at mottaker åpner en pakke |
| Name spoofing - | Bruke en feilstaving for å få trafikk f.eks Phyton istendenfor Python |
| DoS eller DDos | Oversvømme en side med pakker for å hindre den i å kjøre normalt |
| SYN flood | Sende en haug med tilfeldige TCP SYN for å slite ut en mottakers set ac TCP forbindelser |
| Password breaking - | Automatiserte systemer som prøver et passord flere ganger |
| Port scanning - | Prøve å koble til hver protokoll port på en vert for å finne en svakhet |
| Packet interception – vanskeligst, men mest skade potensiale | Fjerne en pakke fra nettet slik at man muliggjør utbytting og muliggjør mellommann angrep |

Data integrity: beskyttelse fra endring  
Data availability: beskyttelse fra DoS er siden tilgjengelig  
Data Confidentiality: beskyttet mot wiretapping?  
Privacy: er senderens identitet skjult?

|  |  |
| --- | --- |
| Teknikk | Grunn |
| Hashing | Data integrity |
| Encryption | Privacy |
| Digital signatures | Meldingsautentisering |
| Digital certificates | Senderautentisering |
| Firewalls – brannmurer | Site integrity |
| Intrusion detection systems | Site integrity |
| Content Scanning & deep packet inspection – filskanning | Site integrity |
| Virtual private Networks (VPNs) | Data confidentiality |

For å kryptere bruker man en: private key og public key  
RSA-kryptering bruker mod-regning og nøkler.   
autentisere med digitalsignatur bruker også nøkler  
En brannmur bruker pakke filtrering for å hindre uønsket kommunikasjon  
Sikkerhetsteknologier: PGP, SSH, SSL, TLS, HTTPS, IPsec, RADIUS og WPA

**Kap. 32 The internet of things**

IoT bruker innebygd system og M2M (machin to machin) for å samhandle.   
Hjemmeautomasjon, smart grid, sikkerhet.  
Lav energiforbruk. Bruker mesh for å slippe å sende sterke signaler.   
Ca. 3 pcer per person i dag.

**PPTer  
Hardware**Joseph Marie Jacquard - Jacquardvev (1801) 🡪 Charles Babbage - Analytical (1840) og Difference (1832) engine 🡪 Herman Hollerith - IBM - hullkort (ca 1880) 🡪 Atanasoff - ABC (ca. 1940) 🡪 Konrad Zuse - Z3 (Berlin, 1941) 🡪 ENIAC (Electronic Numerical Integrator and Calculator - 1946) 🡪 John von Neumann, Alan Turing 🡪 Transistoren (ca 1947), Integrerte kretser (60- tallet), Mikroprosessor (1971 -> ) 🡪 Ivan Sutherland - Sketchpad (1963) 🡪 Douglas Engelbart og SRI (60-tallet) 🡪 Xerox PARC (70-tallet) 🡪 Steven Paul Jobs og Apple Macintosh (1984) 🡪 ARPANet/Internet (1969) 🡪 WWW, HTML (1993) - Tim Berners-Lee 🡪 Bærbart utstyr for lomma (1994/2007) - Newton/iPhone/iPAD

Datamaskiner er overalt - miniatyrisering og masseproduksjon  
Datamaskin: Input/Output-enheter 🡪 Mus, styreflate, mikrofon, kamera, skjerm, printer… Hovedenhet 🡪 (Beregningsenheten – CPU, Hukommelse - RAM ), Eksterne enheter 🡪Harddisker ++

Motherboard eller Hovedkortet inneholder mikroprosessor-chipen, også kalt den sentrale prosessor-enheten (CPU), og minnet

RAM: Random Access Memory – flyktig (ikke permanent) Hva betyr «Random Access» – Alle elementer kan hentes direkte.  
Hva datamaskiner kan – Deterministisk utføre (eksekvere) instruksjoner for å behandle informasjon – Datamaskinen må ha instruksjoner å følge   
Hva datamaskiner ikke kan – Ingen fantasi eller kreativitet – Ingen intuisjon – Forstår ikke ironi, tvetydighet, forhold, sømmelighet, humor – Er ikke hevngjerrige eller onde – Er ikke målbevisst – Har ikke fri vilje – Er ikke som i Hollywoodfilmer: Terminator, Matrix, AI¨

5 grunnleggende deler/subsystemer – Minne – Kontrollenhet – Aritmetisk/logisk enhet (ALU) – Input-enhet – Output-enhet Bundet sammen av – Busser

Forskjellige datatyper tar opp forskjellig plass – char (tar vanligvis opp 1 byte (ASCII) eller 2-4 byte (UNICODE)) – float (8 eller 10 byte) – etc…  
Noe en ALU kan gjøre Foruten ADD, MULT (Multipliser) og DIV (Dividere), så finnes

også andre instruksjoner:

– Flytt bit-ene i et «ord» til venstre eller høyre, fyll de tømte plassene med null,

og kast bort bit-er som detter utenfor enden

– Beregn logisk AND (test om par av bit-er begge er «sann»), og logisk OR

(test om minst en bit i paret er «sann»)

– Test om en bit er null eller ikke-null, og hopp til et nytt sett instruksjoner

avhengig av utfallet

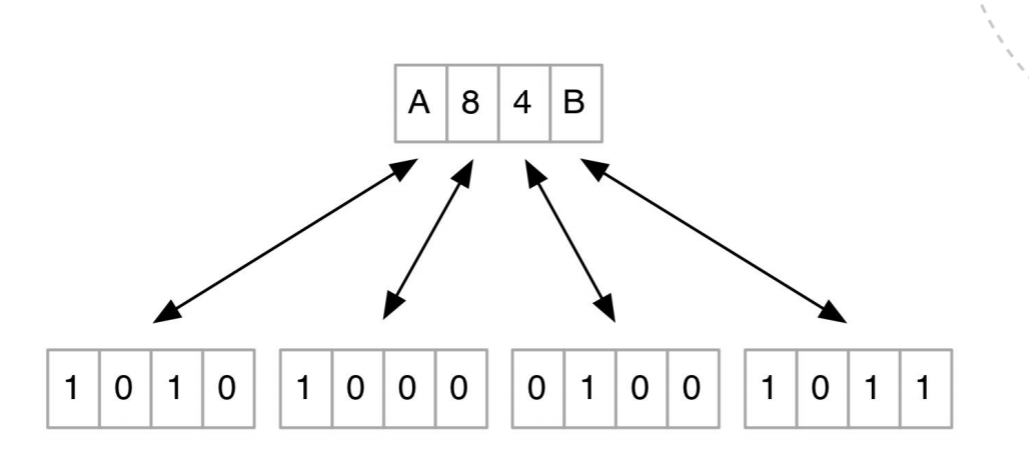
– Flytt informasjon i minnet

– Sjekk signaler fra input/output-enhetene

**PPT digital representasjon**Bits: 0/1

Bytes: 8 bits, 28 = 256 symboler (mønster)

– 0000 0000, 0000 0001, 0000 0010, … , 1111 1110, 1111 1111  
P and A-koding (Present and Absent) – Presence / absence  
Metadata er data om data, det vi trenger for å tolke riktig

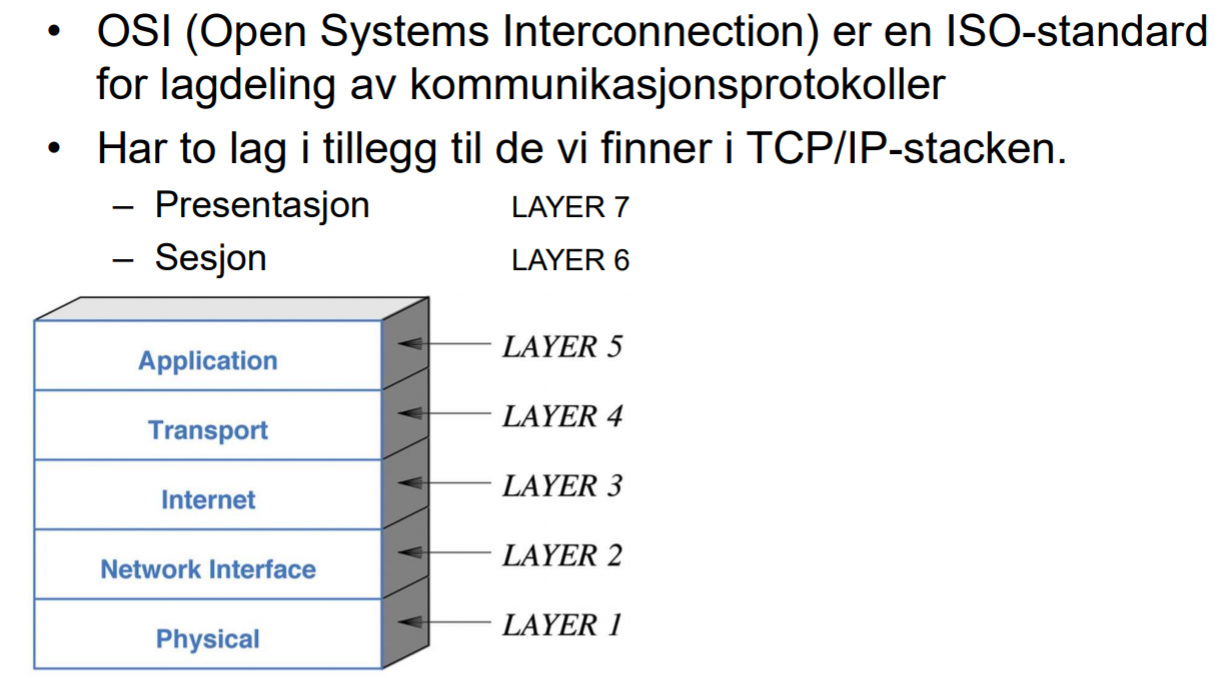
Hex til/fra binært  


Tallene i en byte: 0-127 brukes til positive heltall, 128-256 brukes til negative tall. Får å bytte mellom negativ og positiv. Flipp alle bitsene og legg til 1.  
Digitalisering av lyd: - Et objekt lager lyd ved å vibrere et medium - Kraften eller intensiteten av bølgene avgjør volumet - Frekvensene (antall bølger per sek) avgjør tonehøyden (pitch)  
A til D konvertering - ”Sample” eller ta målinger på bestemte intervall - Antall samples per sekund kalles samplingsfrekvens (rate) - Jo raskere frekvens, jo mer nøyaktig opptak av lyden Vi kan kun få omtrentlige målinger av lyd - Hvis et **ekstra bit vil bli brukt, vil lydsamplet bli dobbelt så nøyaktig** - Flere bits oppløsning gir mer nøyaktig digitalisering - Digital CD lyd representeres ved **16 bit**, dvs. 65 536 forskjellige lydnivåer  
Digital data kan reproduseres nøyaktig. - Kan dette være et problem? Ja

Desimaltall av binært: 101,1011 = 1\*2^3 + 0\*2^2 + 1\*2^0 + 1\*2^(-1) + 0\*2^(-2) + 1\*2^(-3) + 1\*2^(-4)  
For å lagre et flyttall:  
1. Konverter det til et binært flyttall   
2. Konverter det til eksponensialform   
3. Lagre fortegnet som 1 bit   
4. Lagre eksponenten som et heltall   
5. Drop første bit i mantissa og lagre resten (som ‘venstrejustert’ eller null-paddet på høyre side)

Bildefiler: TIFF, PSD, BMP, PNG, GIF, JPEG/JPG  
tapsløs GIF  
tap JPEG, MP3

Printer ut i cyan, magenta, yellow og black CMYK  
Bildekomprimentering ikke endre lysstyrke, men endre til mindre fargenyanser(chrominance)

**Nettverk**Distrubisjonsnett: CATV, Analog radio, DAB/+, DVB-t  
Kommunikasjonsnett: GSM, internett, 4G/LTE, Tetra, PSTN, Maritim radio (VHF)  


Ting som ikke er i PPT eller boken, men tidligere oppgaver:  
Hoffmankode, inkrementell utvikling, vannfallsmetoden, akseptansetesting