Datasystemer 2017.

Datamodellering

Teori og øvingar i denne delen av kurset, veke 2-9, er delvis eksamenspensum, men skal i hovedsak vera grunnlag for miniprosjektet som skal gjennomførast i veke 7-9. Karakteren på miniprosjektet utgjer 40 % av mappekarakteren i faget Datasystemer og nettverk.

Pensumliste til eksamen datamodellering og programmering 2017:

Hansen og Mallhaug: Databaser:

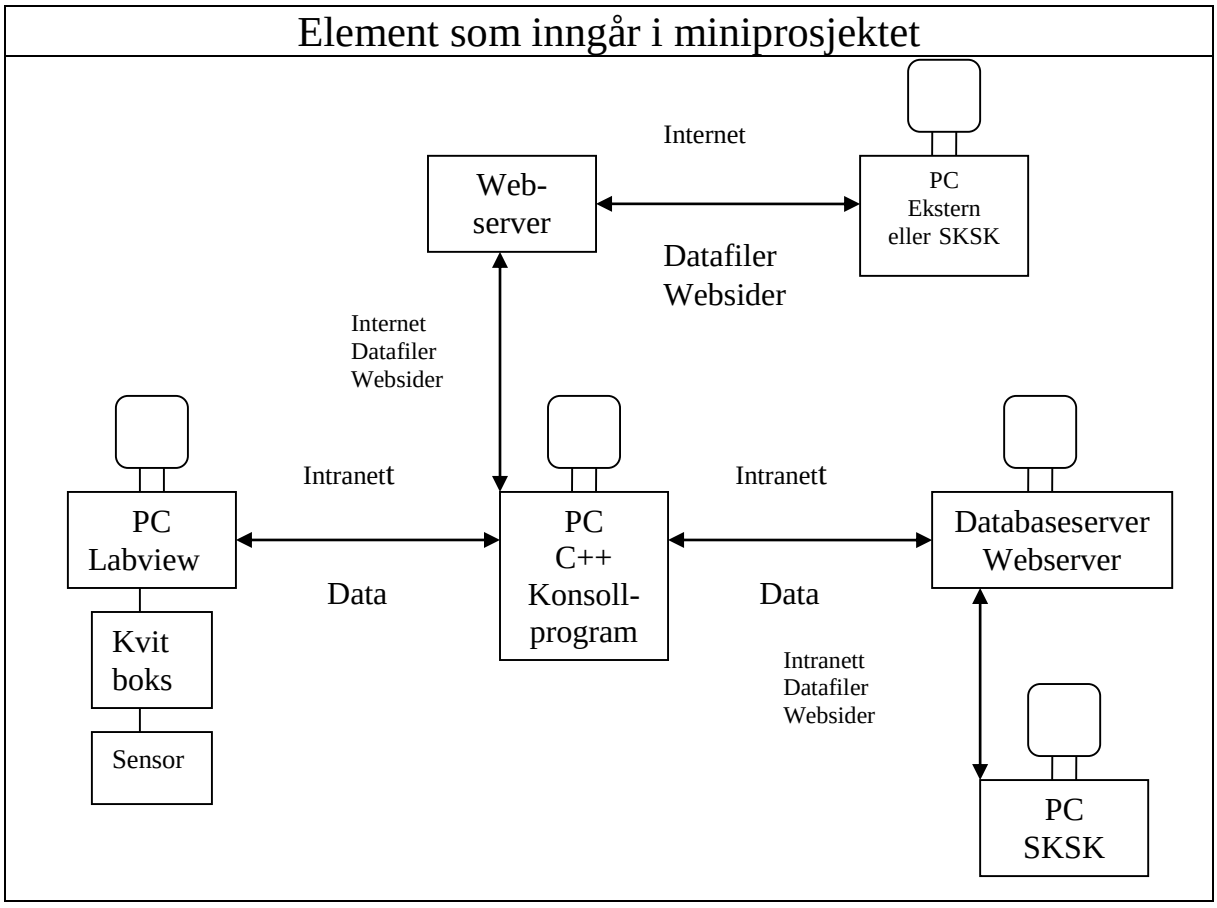
Kap. 1-6

Lervik og Ljosland: Programmering i C++:

Kap. 10, 14, 15.1-15.2, 16.5, 18.1 - 18.4.

Diverse utgitt informasjon i form av hefter og elektroniske dokument.

Pensumlista kan bli endra i løpet av kurset.



Datamodellering 2017

Miniprosjektoppgåve.

Innlevering: Innan onsdag 2. mars
Arbeidsgruppe 2 og 2 eller individuelt .
Evaluering: Karakteren på miniprosjektet utgjer 40 % av mappekarakteren i faget Datasystemer.

Alle skal lage til ein sensor som er kopla til ein PC via ein "kvit boks".
Sensoren skal måla ein fysisk størrelse: Temperatur, trykk, lysstyrke, peilevinkel, avstand, el.l.
I utgangspunktet skal alle sensorane gi ei spenning ut. Dei skal kalibrerast slik at spenninga kan reknast om til fysisk størrelse.

På PC-en skal det liggja ein database som skal innehalde tabellar med følgjande opplysningar om sensorane:
SensorID, Sensornamn, Leverandør, Fysisk størrelse, Måleeining, Kalibreringskonstantar, Operatør, Samplingsintervall, Plassering (maritim plassering:Fartøy, Fartøytype, Maskinrom, bro, kjølevatn, sjø….), ansvarlege offiserer, grad, telefonnr, installeringdato, måletidspunkt, måledata(spenning), start og stopptid for siste måleperiode, opplysningar om PC-en som sensoren ligg på (IP-adresse, brukarnamn og passord med mer) + andre relevante data.
Databasestrukturen skal vera identiske på alle PCane, for vi skal seinare prøve på "replisering", dvs at kvar database oppdaterer ein sentral database som så kan oppdatere alle databasane (nettverksbasert forsvar der kvar eining har meir eller mindre oppdatert oversikt over alle dei andre einingane sine data).(Avsnitt 9.4 i læreboka)
Databasane skal vera normalisert (kapittel 3 i læreboka)
Alle i klassen må vera med på design av databasen.

Kommunikasjon mellom sensor og database skal programmerast i Cpp.

Henting av data frå database skal skje via intranett ved køyring av websider . Alle U-nettmaskiner med IP-adresser 192.168.20.xxx skal kunne hente ut oppdaterte data frå din sensor på ein fornuftig måte.

Oppdadterte datasider skal også lastast opp til ein ekstern server med ftp-adresse **ftp.sksk.no**
Kvar gruppe får tildelt eit brukarnamn og passord. Sidene vil då liggja på http:// sksk.no/bachelor/gruppeX der X er gruppenummer.

Fellesdatabasen og replisering er ikkje ein del av denne oppgåva.

Innlevering: Når eg skal evaluere miniprosjektet, vil eg gå rundt på Datateknisk lab og køyre program og studere datatabase og data. Alle gruppene lagar difor ei bruksanvisning som inneheld minst denne informasjonen:
Presentasjon av sensor, koplingsskjema, angitt inngangar på kvit boks.
Nummer på PC der program og database ligg, passord og brukarnamn for database, IP-adresser og portnummer. URL for intranettsider og internettsider.
Instruks for køyring av program.
Lykke til!

Oppstart Miniprosjekt torsdag 9.2.

I miniprosjektet skal det på den virtuelle maskinen liggja ein database som skal innehalde tabellar med følgjande opplysningar om sensorane:

SensorID, Sensornamn, Leverandør, Fysisk størrelse, Måleeining, Kalibreringskonstantar, Operatør, Samplingsintervall, Plassering (maritim plassering:Fartøy, Fartøytype, Maskinrom, bro, kjølevatn, sjø….), ansvarlege offiserer, grad, telefonnr, installeringdato, måletidspunkt, måledata(spenning), start og stopptid for siste måleperiode, opplysningar om PC-en som sensoren ligg på (IP-adresse, brukarnamn og passord med mer) + andre relevante data.
Databasestrukturen skal vera identiske på alle PCane, for vi skal seinare prøve på ”replisering”, dvs at kvar database oppdaterer ein sentral database som så kan oppdatere alle databasane (nettverksbasert forsvar der kvar eining har meir eller mindre oppdatert oversikt over alle dei andre einingane sine data).(Avsnitt 9.4 i læreboka)
Databasane skal vera normalisert (kapittel 3 i læreboka)
Alle i klassen må vera med på design av databasen.

Første time i Miniprosjektet skal klassen ”designe” databasen som skal bestå av fleire tabellar. Ein del tips til korleis tabellane bør lagast:

Tabellane skal vera normalisert (kapittel 3). For å sjekke det, teikn diagram som viser funksjonelle determineringar. Sjå til at tabellane har primærnøkkel og berre atomiske verdiar (1. normalform), ingen partielle determineringar (2. normalform) og ingen transitive determineringar (3. normalform).

Data som blir oppdater ofte (f. eks. måleverdi og måletidspunkt) bør vera i ein annan tabell enn data som berre skjeldan skal oppdaterast (kalibreringskonstantar, starttid, stopptid, namn, PC-nummer osv). Lag eigne tabellar for data som logisk høyrer saman og bruk framandnøklar for å binde saman tabellane.

Når du skal lage programsystem i C++ , kan det vera aktuelt å lage det som fleire småprogram der kvart program skal oppdater ein eller fleire tabellar eller utføre andre oppgåver.

Unngå ”like namn” på felt i samme tabell (postnr og poststad er litt for lik, C++ kan forveksle dei, skriv heller postnr og stad)
Ungå norske bokstavar i feltnamn (C++ skal lage variablar med samme namn så for eksempel ”måleverdi” er problematisk.)

Ta i bruk din virtuelle maskin

NB NB NB

Du logger deg på din virtuelle maskin som administrator, det betyr at konsekvensene av uoverveide handlinger kan være betydelig. Det er f eks ikke lurt å skifte administratorpassord for deretter å glemme det.

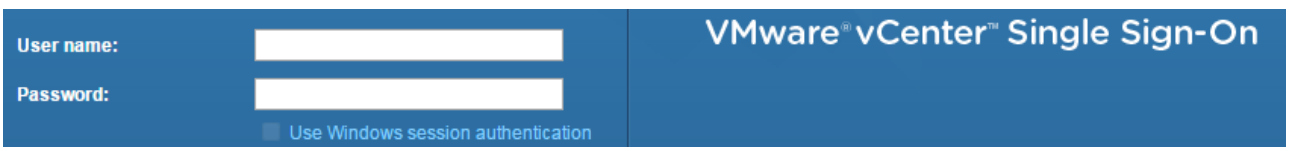
Med uttrykket "fysisk maskin" mener jeg den vanlige datamaskinen du sitter ved, hvor du bruker din vanlige SKSK brukerkonto.

NB NB NB

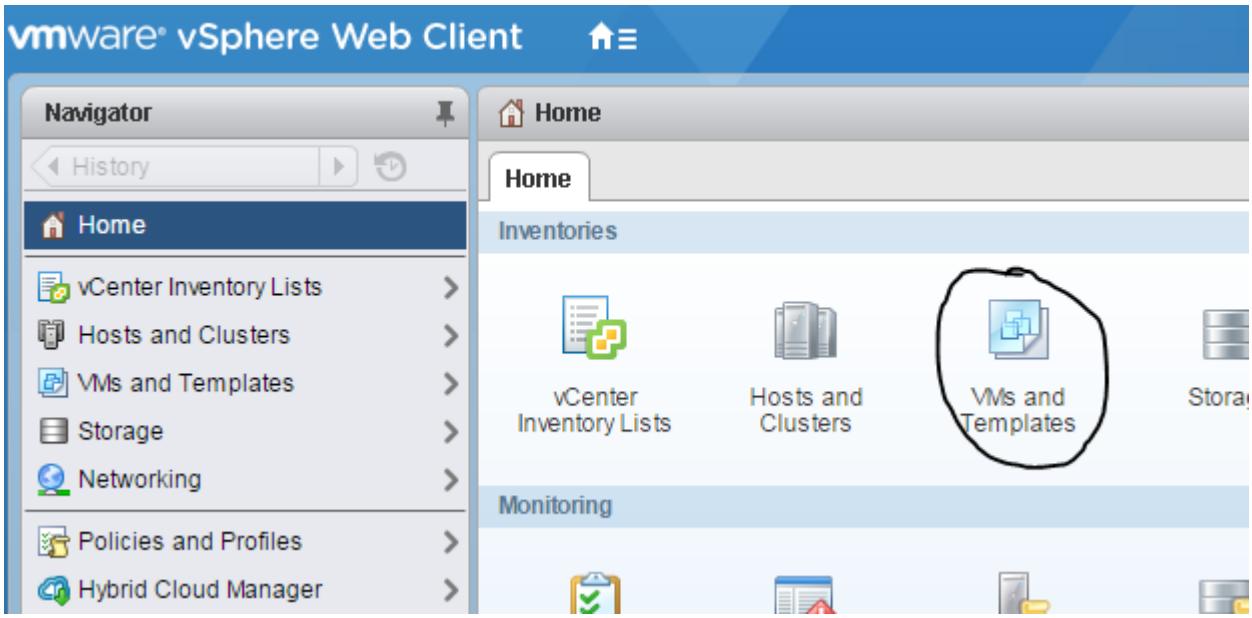
Din virtuelle maskin kjøres på en av skolens tjenere. Vi styrer den virtuelle maskinen gjennom et program kalt `VMware Remote Console`, det skal være installert på skolens ugraderte maskiner. Du må også bruke en Web klient, for øyeblikket er det `Chrome` (testet) og `Firefox` (ikke testet) som er mulighetene. (`Internet Explorer` kan ikke brukes pga et sertifikatproblem som IE ikke tillater deg å overse.)

Du kan også klare deg uten `Remote Console`, men da får du noen irriterende problemer med norsk tastatur. Se side 4 , alternativ startprosedyre.

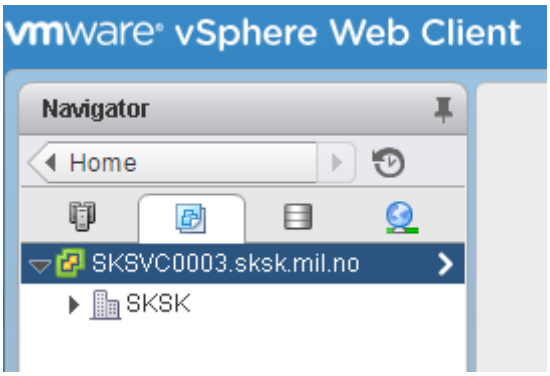
1. Kontakt den tjeneren hvor de virtuelle maskinene ligger: Bruk `Chrome` med følgende adresse: https://sksvc0003:9443/vsphere-client .
2. Bruk din vanlige ugraderte konto og passord til å logge deg på.

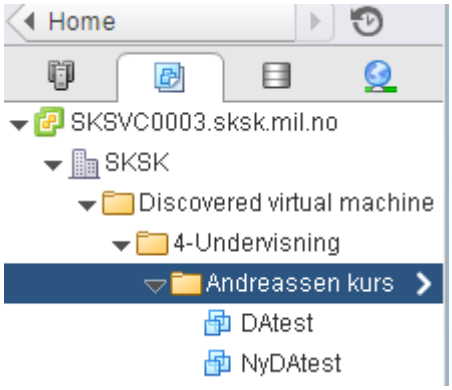


3. Du får opp skjermbildet under, klikk på `VMs and Templates`.



4. I startbildet under (venstre bilde) arbeider du deg frem til folderen *Tveita* `kurs` (høyre bilde), der finner du en virtuell maskin med ditt navn.

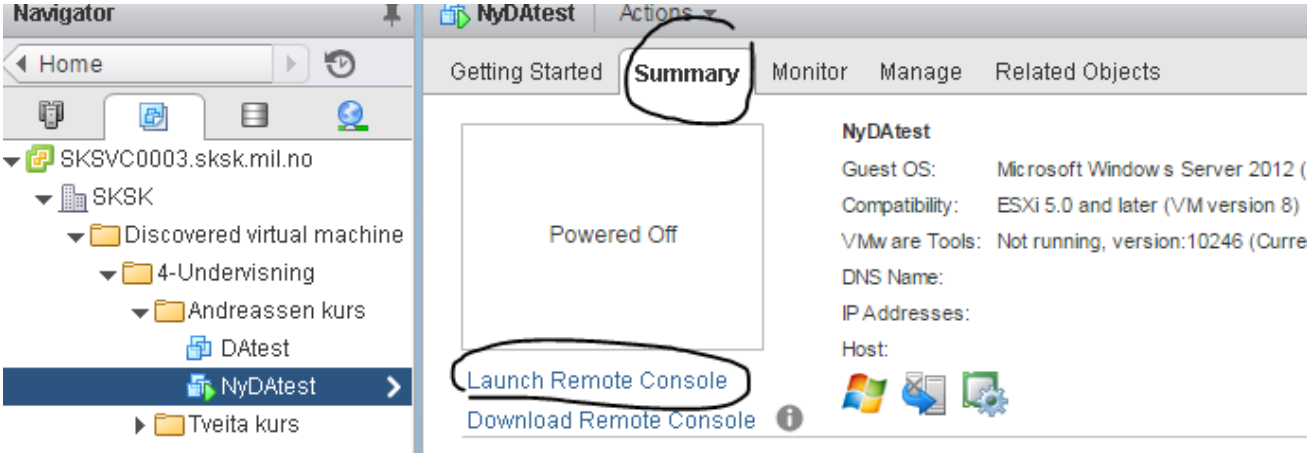




5. Høyreklikk én gang på den, velg `Power` og `Power on`. Hvis du følger godt med, ser du at ikonet for din virtuelle maskin får en grønn pil; din maskin har startet.

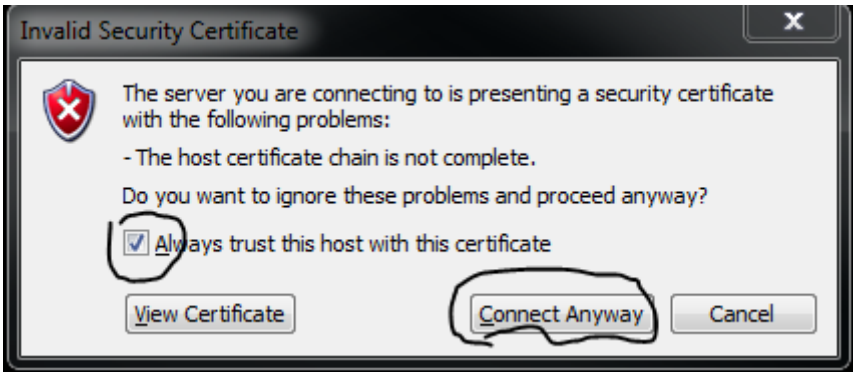
Maskinen din er i gang, men du må åpne et vindu inn til den for å kunne kommunisere med den.

6. Velg arkfanen `Summary` og klikk på `Launch Remote Console`.



Et tomt vindu åpnes, etter noen sekunder dukker VMware Remote Console opp.

7. Du får opp en advarsel om problemer med sertifikatet på den tjeneren som kjører din virtuelle maskin. Velg `Always trust ..` og klikk på `Connect anyway`.



8. Etter noen sekunder ser du standard påloggingsbilde

NB NB NB

Nå har du startet en virtuell maskin. Den oppfører seg som en fysisk maskin, med 1 unntak: **Dersom du ønsker å sende tastesekvensen Ctrl-Alt-Del til din virtuelle maskin, må du i stedet bruke Ctrl-Alt-Insert.** Ctrl-Alt-Del går til din fysiske maskin.

Du er administrator på din virtuelle maskin, det betyr at tabber kan ha interessante konsekvenser.

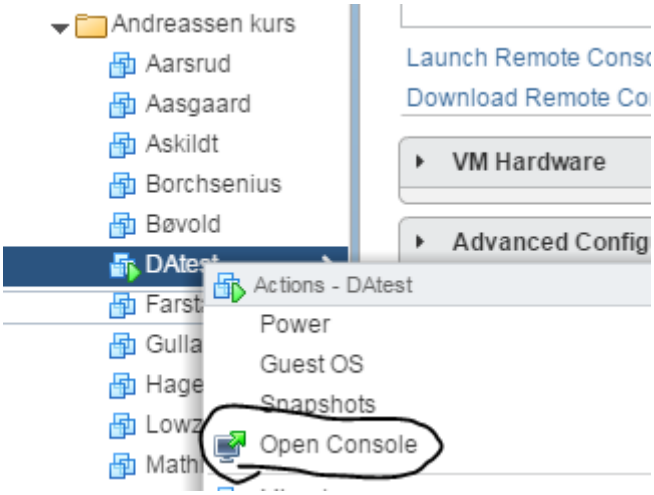
NB NB NB

Alternativ startprosedyre

Det er problemer med å bruke Remote Console. Hvis problemene ikke er løst før vi begynner med øvelser, må vi kjøre følgende alternative prosedyre. Det er punkt 6 som må endres.

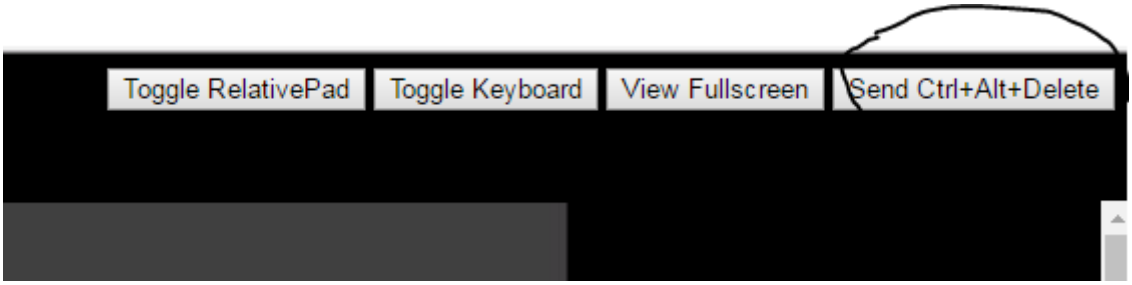
Alternativt punkt 6:

Etter at du har utført «`Power on`»-prosedyren, høyreklikker du en gang til på din virtuelle maskin og velger `Open Console`.



Du får opp et standard påloggingsbilde. Nå vil alt virke som beskrevet, med to unntak:

- `Ctrl-Alt-Insert` virker ikke som beskrevet, du må bruke knappen oppe i øverste høyre hjørne

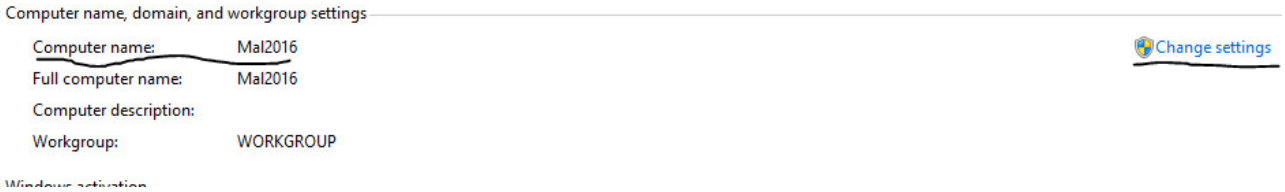


- Norsk tastatur virker ikke.
    - Kan bli et problem hvis du lager avanserte passord.
    - Blir et lite problem i en øvelse på slutten av semesteret, jeg lager oversettingskart hvis nødvendig.

Klikk ett eller annet sted på skjermen til den virtuelle maskinen og logg deg på. Brukernavnet er `Administrator`, passordet er `Tull1234`. **Husk å bruke Ctrl-Alt-Insert!!!!**

Til slutt skal du endre navnet på maskinen, det er litt upraktisk at alle heter det samme. Velg et navn selv, det kan være lurt å unngå nordiske bokstaver.

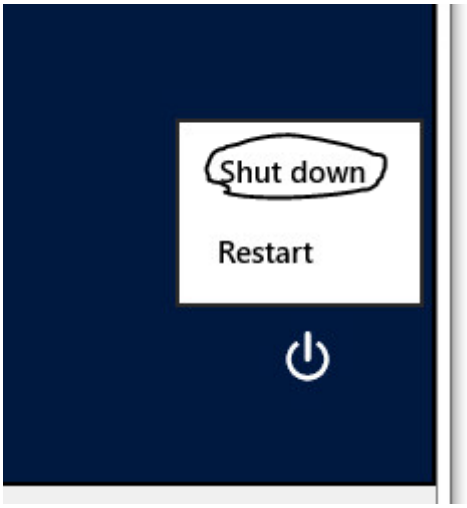
9. `Start→Control Panel→System`, velg `Change Settings` og klikk på knappen `Change`.



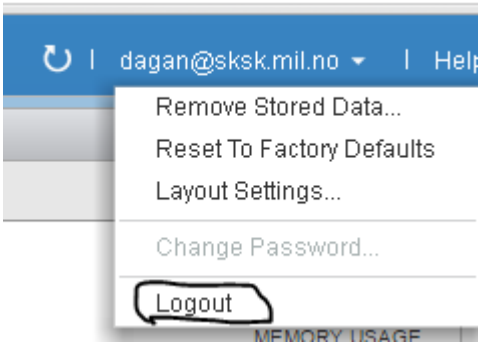
   Skriv inn ditt valg i `Computer name` og klikk på `OK`.

   Som du ser, må maskinen startes på ny etter denne operasjonen.

10. Steng ned den virtuelle maskinen din, husk `Power Off`. Lukk konsoll og konsollvindu. Nå skal ikonet for din maskin ikke lenger være grønt.



11. Logg ut fra tjeneren.



Datamodellering 2017.

Installering av operativsystem og SQL Server .

Vi skal bruke ein virtuell maskin med ferdig installert Windows Server 2012, SQLServer databaseprogram, Visual Studio 2010 og FileZilla ftp-program. Den virtuelle maskinen køyrer på ein server og vi styrer den gjennom eit konsollvindu i programmet WMware wSphere Client. Du loggar inn der med ditt eige brukarnamn og passord. Du får tildelt ein virtuell maskin som du startar opp (dersom han ikkje alt går).

Maskinen din ligg i mappa Tveita kurs og har ditt etternamn og første bokstav i fornamn som namn i katalogen. Men alle maskinane har samme **maskin-namn. Det skal vi endre seinare.**

Opn eit konsollvindu for maskinen og logg deg inn. Passord for maskinen er *Tull1234*. Gjer deg kjent med kva program som ligg tilgjengeleg i konsollvinduet. Høgreklikk for å få fram alle program. Start C++ i Visual Studio 2010 og lag eit lite testprogram for å sjekke kor raskt systemet er.

Sjekk kva katalogar som ligg på C-disken på maskinen

Sjølv om du lukkar konsollvinduet, vil maskinen halde fram med oppgåvene sine.

Seinare skal vi også bruke Labview-programmet på vertsmaskinen i samarbeid med C++ på den virtuelle maskinen.

Installer Internet Information Server (IIS):
Dersom du ikkje finn katalogen Inetpub på C-disken, er ikkje IIS installert. Då må du installere den:

1. På Server 2012 er ikkje alle roller konfigurert. Frå konsollvinduet for den virtuelle maskinen din startar du **Server Manager**. Frå Dashboard vel du **Add roles and features.** Velg **Web Server (IIS).** Under **Role Services:** Kryss også av for **FTP Server.** Velg til slutt **Install.** Når installasjonen er ferdig, velg **close**.
2. På C-disken på maskinen er det no laga ein katalog **inetpub** med underkatalogar **wwwroot** og **ftproot**. Dei skal brukast til å leggja internettsider på og filer som skal overførast ved ftp (File Transfer Protocol).

Datamodellering 2017.

Bruk av HTTP- og FTP-protokoll.

- **FTP-program (FilZilla)**
- **FTP-script**
- **C++ - program**

Når FTP-server og webserver i Internet Information Server er aktivisert, kan vi utveksle dokument mellom datamaskiner ved hjelp av web-leser (f. eks. Internet Explorer) eller FTP-program (f. eks. FilZilla) . Dokument som ligg på hhv wwwroot og ftproot og underkatalogar kan då hentast ned frå server og studerast eller lagrast. **Windows-brannmuren må først slåast av.** Bruk kontrollpanelet til det.

FTP (File Transfer Protocole) -kommunikasjon.

For at andre maskiner skal kunne laste opp og ned filer til din maskin vha FTP-protokoll, må dei oppgi brukernamn og passord til ein katalog på din maskin. Slik lagar du brukernamn og passord til ein katalog:

1. Lag til ein katalog C:\\Inetpub\ftproot**Opplasting** og tilordn brukernamn: ftp og passord: Password1 til den slik:
   Gå til Administrative Tools/Computer Management/Local Users and Groups. Høgreklikk på Users og velg New User. Velg User name: ftp (dersom den alt eksisterer, så lag ftp2) og passord: Password1, fjern avkryssing på "User must change password.." og kryss av for "User cannot change password" og for "Password never expires". Klikk Create og så Close.

2. Lag ei ny gruppe Ftpgruppe (dersom den alt eksisterer, så lag ei Ftpgruppe2): Høgreklikk på Groups og velg New Group. Skriv inn namn Ftpgruppe og klikk Add. Under Select Users klikk Advanced og Find Now. Du får opp ei liste over brukarar. Velg ftp og klikk OK tilbake til New Group. Der klikkar du Create og Close.

3. Gå til mappa **ftproot\Opplsasting** i Min Datamaskin, høgreklikk på mappa og velg Properties/Security. Gi gruppa Ftpgruppe rettigheter: Read, Write og List Folder Contents. Du må kanskje først bruke Add.. for å leggja til Ftpgruppe som gruppe.

4. Vi skal tillate andre brukarar å skrive til mappa ftproot:
   Gå til Administartative Tools/Internet Information Services(IIS) Manager. Ekspander **Datamaskinnamn** . Høgreklikk på **Sites** og velg **Add FTP Site**. Velg FTP site name: **ftp**. Velg Physical path **C:\inetpub\ftproot**. Velg **No SSL** og Next. Kryss av for **Authentication: Basic**. Allow access to: **Datamaskinnamn\ftp** og gi Permission **Read** og **Write**. Finish.

5. Gå saman to og to og prøv å laste opp og ned filer til mappa **Opplasting** ved å bruke programmet **FilZilla .** Bruk user_id: **Datamaskinnamn\ ftp** og Password: Password1. Bruk **ipconfig i Command Prompt** for å finne IP-adresser (IPv4 Adress) 192.168.20.xx der xx er eit tal mellom 20 og 255. Noter deg IP-adressa di.

FTP-script

Når slik opplasting og nedlasting skal automatiserast, må vi lage eigne skript eller program som brukar http- eller ftp-protokoll til opplasting og nedlasting.

Eit ftp-script kan brukast direkte i cmd- eller DOS-vinduet (RUN/cmd). Eksempel:

Oppgåve 1

Lag ei lita fil mittnamn.TXT som du skal sende til ein nabo.
Lag ei skriptfil i notisbok FTPOPPL.SCR med dette innhald:
Open 192.168.yyy.xxx
brukarID
password
cd Opplasting
put mittnamn.TXT
quit

Lagre begge filene i My Documents.

Avtal med ein makker om å utveksle IP-adresse, brukarID og passord.
I CMD-vinduet skiftar du til katalog My Documents og skriv inn:

ftp –s:FTPOPPL.SCR

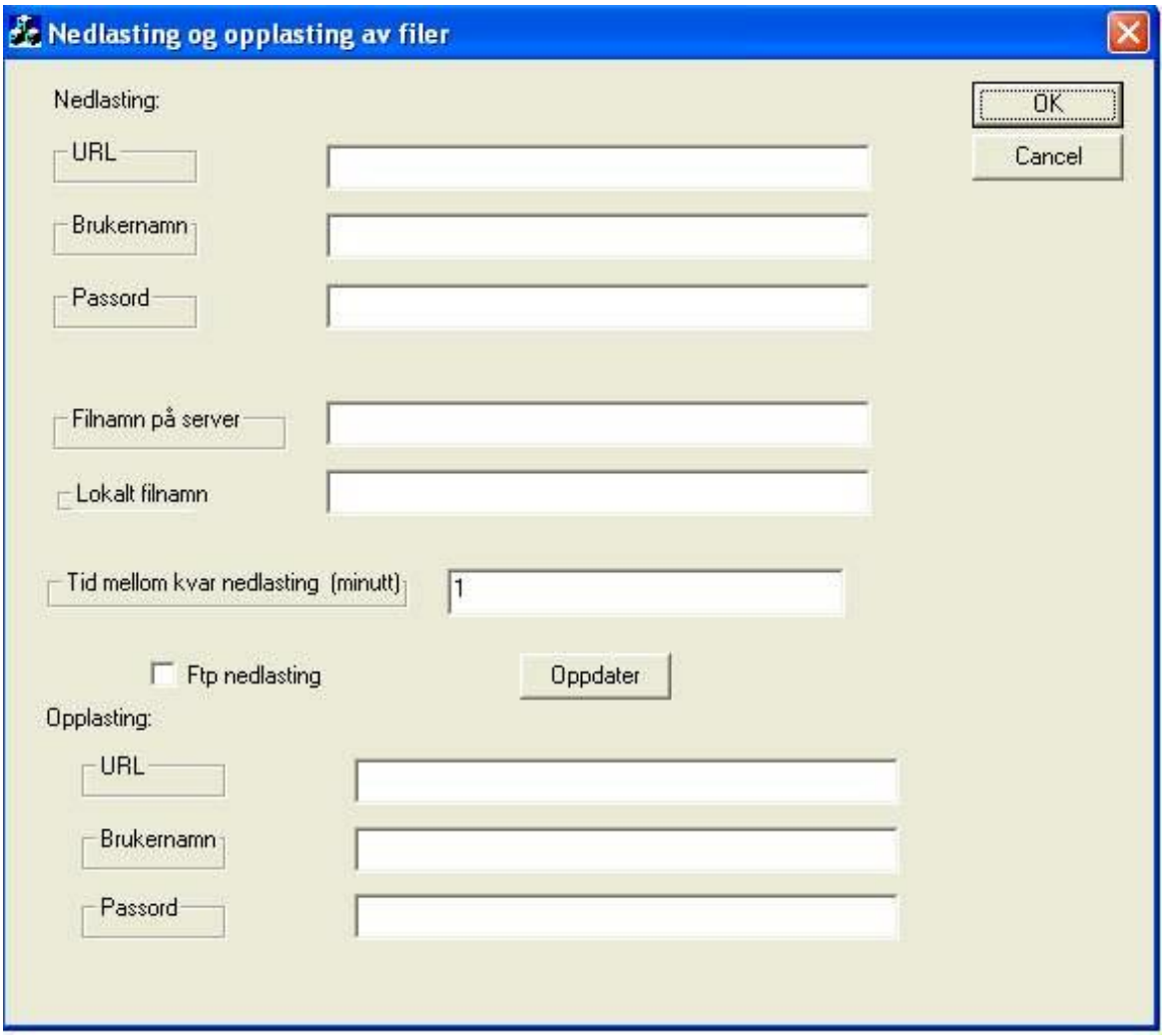
og fila mittnamn.TXT blir lasta opp til katalogen Opplasting på ftproot-katalogen på maskinen med URL 192.168.yyy.xxx.

Andre FTP-kommandoar finn du ved å skrive (i cmd-vinduet): ftp ?

Http- og ftp-funksjonar i C++

Vi skal så lage eit Visual Cpp-program **httpftp** som kvar time lastar ned ei fil frå internett og lastar samme fila opp til ein ftp-server. Vi skal kunne velje mellom http-nedlasting og ftp-nedlasting og bruke eit ftp-skript til opplastinga. Funksjonane **httpGetFile** og **ftpGetFile** ligg på fila **CRobotInternet.cpp.** Funksjonane er Public funksjonar i klassen **CRobotInternet** som er definert på fila **CRobotInternet.h.**
Filene kan lastast ned frå http://sksk.no/Tveita/Program/Datasystemer.htm . Ekstraher filene og legg dei førebels på ein katalog C:\CPP\ . Studer klassen og noter deg kva som er inndata og utdata til dei to funksjonane ftpGetFile() og httpGetFile() som vi skal bruke.

Nytt i det programmet som vi skal lage, er også bruk av Timer.
Slik skal applikasjonen sjå ut når han er ferdig:



6. Start Visual Studio 2010 (Visual C++) på den virtuelle maskinen og velg File/New/Project/MFC Application og velg prosjektnamn **httpftp.** Velg lagring på **C:\Cpp.**
7. Velg **Dialogbased** og **fjern eventuell avkryssing på Use Unicode libraries**. Velg default på resten av step 1 to 6.
8. **Build** og **Start Without Debugging** for å sjå at alt er i orden. Fjern knappar frå formen.
9. Bruk Toolbox (View Toolbox) til å lage input-felt for URL, Brukernamn, Passord, Filnamn (på server),Lokaltfilnamn og Tid(mellom kvar nedlasting).
   Velg Id på felta: IDC_URL, IDC_Bruker, IDC_Passord, IDC_Filnamn, IDC_Lokaltfilnamn og IDC_Tid.
10. Lag ein check-boks for ftp-nedlasting (for å velja mellom ftp og http). Velg IDC_ftp
11. Høgreklikk i eitt av felta og velg ClassWizzard/Member Variables og tilordn variablar m_URL, m_Bruker, m_Passord, m_Filnamn, m_Lokaltfilnamn (alle som Cstring) og m_Tid som int (velg min=1 og max=60) og m_ftp som BOOL.
12. Lag ein Knapp som du gir Id: IDC_Oppdater og som du skriv Oppdater på. Velg View/ClassWizzard/Commands Object Id = IDC_Oppdater og Message BN_CLICKED Add Handler . Godta OnClickedOppdater som funksjonsnamn og velg Edit Code. Skriv inn setningar her som får dataene til å gå frå felta i skjemaet og inn i variablane og evt. endre Timer-innstilling (kan også kopiere teksten frå http://sksk.no/Tveita/ProTull1234gram/httpogftpiCpp.txt) :

```
int GamalTid=m_Tid;
UpdateData(TRUE);  //Data frå skjema til variablar
if(GamalTid!=m_Tid){   // dersom du har endra intervall mellom nedlastingar
        KillTimer(1);
        int millisekund=m_Tid*60000;
        int installResult=SetTimer(1,millisekund,NULL);
        if(installResult==0)MessageBox("Kan ikkje installere timer");
}
```

13. Når programmet startar, blir det køyrt ein funksjon som heiter OnInitDialog(). I den funksjonen startar vi ein timer som vil køyre ein funksjon OnTimer() med jamne mellomrom (førebels 1 minutt). Velg View/ClassWizzard/Message Maps Object Id = CHttpftpDialog og Message=WM_INITDIALOG. Velg Edit Code, og etter
    // *TODO: Add extra initialization here* Skriv inn programkode:

```
int installResult;
installResult=SetTimer(1,60000,NULL); //60000 millisekund
if(installResult==0)MessageBox("Kan ikkje installere timer");
m_Tid=1; //Eitt minutt
UpdateData(FALSE); //Oppdaterer skjemafelt
```

14. Når programmet avsluttar, må vi fjerne timeren frå datamaskinen: Velg View/ClassWizzard/Message Maps Object Id = CHttpftpDlg og Message=WM_DESTROY. Add Handler. Velg Edit Code, og etter
    // *TODO: Add your message handler here* Skriv inn programkode:

```
KillTimer(1);
```

15. Så skal vi lage funksjonen OnTimer() : Velg ClassWizzard/Message Maps Object Id = CHttpftpDialog og Message=WM_TIMER. Velg Add Handler, godta namnet OnTimer() og velg Edit Code og etter linja
// *TODO: Add your message handler here* Skriv inn programkode for alt som skal skje kvar gang tida er inne:

```
UpdateData(TRUE);  //Data frå skjema til variablar
int Resultat=0;
if(m_ftp){        //Dersom vi skal bruke ftp-nedlasting
CString URL=m_Bruker+":"+m_Passord+"@"+m_URL;NULL;
CString Dir="";         //Katalog på server(tom)
CString Feilmelding="";
Robot.ftpGetFile(URL,Dir,m_Filnamn,m_Lokaltfilnamn,Resultat,Feilmelding)
if(Resultat!=0)MessageBox(Feilmelding);
}
else              //http-nedlasting
        {         CString URL=m_URL+"/"+m_Filnamn;
                  CString Feilmelding="";
                  Robot.httpGetFile(URL,m_Lokaltfilnamn,Resultat,Feilmelding);
                  if(Resultat!=0)MessageBox(Feilmelding);
        }
```

16. Kopier filene **CRobot.h, CRobotInternet.h** og **CRobotInternet.cpp** inn i katalogen der programmet dit ligg.
Legg inn ein setning
#include "CRobotInternet.cpp"
på toppen av fila før #ifdef _DEBUG.
Du må fjerne setninga #include <stdafx.h> frå fila CRobotInternet.cpp for ikkje å få dobbelt opp med den include-setninga.
Etter #endif direktivet på toppen av fila, deklarerer du så
CRobotInternet Robot ; // Robot er eit objekt av klassen CRobotInternet

17. Kompiler og køyr programmet så langt. Prøv å laste ned web-sida http://sksk.no/index.htm Det er ikkje nødvendig med brukarnamn og passord ved http-nedlasting. Sjekk at du har fått lasta ned fila. Prøv å laste ned (ftp-nedlasting) ei fil frå ftp-katalogen på din eigen virtuelle maskin 127.0.0.1.

18. For opplastinga kan vi velja mellom å bruke funksjonen Robot.ftpPutFile() eller å bruke ftp-script. I punkt 20 og 21 lagar vi eit ftp-skript og brukar det. I begge tilfelle må vi få lagt inn IP-adresse, brukarnamn og passord for server som skal motta fila. Legg inn dei tre siste felta for **opplasting** av fil. Du kan kopiere dei 3 øverste og gi dei namn IDC_URL2, IDC_Bruker2 og IDC_Passord2 og tilordn variable til alle: m_URL2, m_Bruker2 og m_Passord2.

19. Default opplasting skal vera til lærarmaskinen sin fpt-katalog. Under CHttpftpDlg::OnInitDialog() etter installeringskoden for Timer legg du inn litt ekstra kode for default-adresser m.m. :
    m_Tid=1;//Default tid mellom kvar ned- og opplasting er 1 minutt
    m_URL="sksk.no"; //Default hentar vi index.htm frå sksk.no ved httpGetfile()
    m_Filnamn="index.htm";
    m_Lokaltfilnamn="dittnamn.htm";
    m_URL2="192.168.20.37"; //Virtuell LærarPC
    m_Bruker2="Test1/ftp"; //Lagt inn som USER på mappe //Inetpub/ftproot
    m_Passord2="Password1"; //Lagt inn som passord for bruker ftp
    *UpdateData……*osv som før

20. Kvar gang vi trykker på Oppdater-knappen, skal maskinen lage eit nytt opplastings-script: Under CHttpftpDlg::OnClickedOppdater() etter oppdateringskoden for Timer:
    ofstream Utfil; //Lagar opplastingsskript-fila
    Utfil.open("ftpoppl.scr");
    Utfil<<"open "<<m_URL2<<endl;
    Utfil<<m_Bruker2<<endl;
    Utfil<<m_Passord2<<endl;
    Utfil<<"put "<<m_Lokaltfilnamn<<endl;
    Utfil<<"quit"<<endl;
    Utfil.close();
    Husk **#include <fstream>** etter dei andre include-filene på toppen av programmet og **using namespace std;**

21. Under CHttpftpDlg::OnTimer(UINT nIDEvent) etter koden for nedlasting:
    if(Resultat==0){ //Køyrer opplastingsskript
    system("ftp -s:ftpoppl.scr");
    }

22. Kompiler og køyr programmet. Prøv å laste ned ymse filer frå både web-server og frå ftp-server. De får utlevert brukernamn og passord til mapper på ftp-server på SKSK.NO. Bruk FileZilla til å laste opp filer til denne serveren, og bruk programmet som de no har laga, til å laste filene ned og lagre dei lokalt.

Etablering av database *Fubase*

Denne databasen blir brukt som eksempel i læreboka . Databasen er presentert i figur 4.2-4.3 i boka. Alle data som skal inn, er vist på side 107. **Du skal gå gjennom alle eksempel i kapittel 4.2 og kapittel 5, etablere alle tabellane side 107 med data og nøklar.**

1. Start SQL Server Management Studio. Velg Database Engine, Datamaskinnamn\DB, Windows Authentication og klikk Connect. (Frå SQL Server Management Studio kan vi administrere fleire databaseserverar og tenester. Det skal vi gjere seinare.)
2. Ekspander **Databases** og ekspander **Fubase2** som er ein ferdig database (sjå boka). Ekspander **Tables**, høgreklikk på nokre av tabellane og velg **Edit Top 200 Rows** eller **Select Top 1000 Rows.** Vi skal no lage ein ny database med slike tabellar.
3. Høgreklikk på Databases og velg New Database
4. Gi databasen namnet *Fubase* og bruk default settingar. Klikk OK. Det blir oppretta to filer Fubase.mdf og Fubase_log.ldf. Den første vil innehalde alle data, den andre vil logge alle endringar som vi gjer. Vi kan dermed "spole tilbake" om vi vil kanselere endringar.
5. Klikk på ikonet for Neww Query. Dette er programsystem for å utforme og køyre spørjingar mot databaser.
6. Skriv: USE Fubase; og klikk EXECUTE (!) eller bruk rullegardinmeny for å velja Fubase.
7. Skriv inn eksempel på side 108 og køyr det for å opprette ein tabelltabell. Skriv inn eksempel side 116 og 117 for å fylle data i tabellen.

8. Lag og fyll ut tabellar vidare ved å bruke SQL Server Management Studio: Ekspander Databases/Fubase. Høgreklikk på Tables og velg New Table. Fyll ut kolonne-namn, datatype og kolonne eigenskapar (her kan du for eksempel velja Identity). Når du har laga alle kolonnene, klikk på Lagre og oppgi tabellnamn.
9. Alle tabellar som er lagra, dukkar opp i Object Explorer-vinduet under Tables. Dei får gjerne eit prefiks dbo. Som står for data base owner. Når du skal referere til tabellen i ei spørjing, må du av og til oppgi fullt namn på tabellen.
10. Det blir laga eit QUERY skript også når tabellen blir laga i Management Studio. Dersom du vil sjå skriptet, kan du høgreklikke på tabellen og velja Script Table As og velja CREATE to New Query Editor Window.
11. Du kan leggja inn data også frå SQL Server Management Studio: Høgreklikk på ein tabell og velg **Edit Top 200 Rows**. Så kan du skrive inn data. Legg merke til at du ikkje får skrive inn data i Identity-felt (COUNTER), det feltet blir oppdatert automatisk. Når du er ferdig med innskriving av data, skal du **ikkje** lagre. Dataene er automatisk på plass i databasen, men av og til må du ta ein Refresh. Høgreklikk på databasen og velg Refresh.
12. Du kan også importere alle tabellane i Fubase frå ei EXCEL-fil som ligg på SKSK.NO: Fubasebackupnov2007. Last ned denne til skrivebordet på den virtuelle maskinen. I SQL Server Management Studio: Høgreklikk på database Fubase, velg Task/Import data, velg DataSource Microsoft Excel, velg Destination SQL Native Client og database Fubase, vidare vel du Copy Data from Tables og kryssar så av for dei tabellane du vil kopiere (ikkje velg dei med $-teikn på). Du må i forkant av dette ha endra namn på tabellar som du har laga før. Du må før kopieringa sjekke at kolonnene i dine nye tabellar er av rett type : Velg ein tabell og Edit Mappings. Her kan du endre på dataformat.

Datamanipulering.

Når databasen er etablert med alle data, skal du gå gjennom alle eksempel i kapittel 4.3-4.4 og kapittel 5..
Lag også eigne spørjingar.
No er det på tide å starte opp med oppgåvene i innlevering 1.

Innlevering Datamodelering til fredag 03.02.2017

I løpet av fredag 03.02.17 skal databasen FUBASE vera komplett med alle data frå læreboka.

Innleveringa består av eitt SQL-skript og utskrifter av responsen for kvar del.

Lag eitt stort SQL-skript for alle desse oppgåvene(alle oppgåvene skal utførast med skript, ikkje med direkte innskriving i tabellar). Alle skripta må kunne køyrast med utgangspunkt i tabellane i den originale Fubase-databasen:

1. Legg inn 5000 Bergen i tabell Sted.
2. Lag ein tabell **ny_Student** som er identisk med tabell **Student**
3. Endre semester H97 til H10 og V98 til V11 i tabell Fagvalg.
4. Lag ei liste over fag og eksamensdato alfabetisk sortert etter fagnamn.
5. Lag ei liste med namnet til studentar som har minst ein karakter betre enn 2.0. Kvar student skal berre forekoma ein gang på lista. Lista skal vera sortert alfabetisk etter namnet på studentane.
6. Lag ei liste over fag som har DATA i fagnamnet.
7. Lag ei liste over namnet på studentar som bur samme plass som eksamensskulen er.
8. Telj opp kor mange studentar som bur samme plass som eksamensskulen er.
9. Finn på ei oppgåve sjølv der det er naturleg å bruke kommando ”INNER JOIN”, og lag skript til oppgåva.
10. Finn på ei oppgåve sjølv der det er naturleg å bruke kommando ”UNION”, og lag skript til oppgåva.
11. Finn på ei oppgåve sjølv der det er naturleg å bruke kommando ”IN”, og lag skript til oppgåva.
12. Finn på ei oppgåve sjølv der det er naturleg å bruke kommando ”CREATE VIEW”, og lag skript til oppgåva.

Skriptet skal prøvast ut, og responsen kan du skrive inn (klipp og lim) som komentar under skripta.

Legg inn /* før og */ etter komentar.

Skriptet lagrar du i ei mappe på datamaskinen: Datamodelering innlev 1, og kopier den mappa inn i din katalog på Innlevering Tveita.

Ha ei god arbeidsveke!

Datamodellering 2017.

SQL-skript i Cmd-vindu

På ein server kan vi køyre eit program **sqlcmd** i Cmd-vindu.

1. Lag ein katalog C:\Filer
2. Bruk SQL Server Management Studio til å lage ei SQL-spørjing f. eks.

   USE Fubase
   SELECT * FROM dbo.sted

   Lagre denne som SQLQuery1.sql i katalogen C:\Filer

3. Når vi skal kommunisere med databaseserveren utanfrå, må vi logge oss inn på den. Vi etablerer difor ein ny Login: Start Microsoft SQL Server Management Studio. Ekspander Security/Logins. Studer kven som har innloggingsrettar. Dersom du ikkje alt har laga **Loginn testbruker1**:
   Høgreklikk på Logins og velg New Login. Velg Login Name: **testbruker1**. Velg SQL Server authentication og Password: **Sjekk1234**. Fjern avkryssing i Enforce password expiration. Velg Default Database: Fubase. Klikk på User Mapping og kryss av for Fubase, db_datawriter, db_datareader, db_owner og public. Klikk OK. Ekspander Databases/Fubase/Security/Users. Sjekk at testbruker1 er komen med som User.
   For å sjekke om du kan logge inn med **testbruker1**, kan du avslutte Microsoft SQL Server Management Studio og prøve å logge på igjen med **SQL Server Authentication** Og brukernamn **testbruker1.**

4. Gå til Cmd-vinduet og skift katalog til C:\Filer. Skriv sqlcmd -? For å få ein oversikt over alle komandoar som du kan bruke i **sqlcmd.**

5. Lag ei batch-fil **Database.bat** i Notepad og lagre den i C:\Filer:

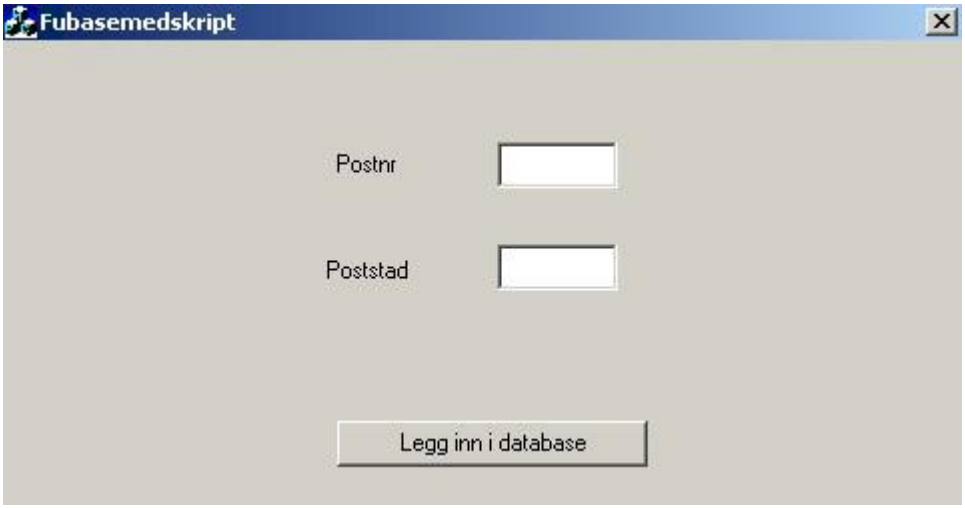
   Sqlcmd –S Datamaskinnamn\DB -U testbruker1 -P Sjekk1234 -l 100 -i C:\Filer\SQLQuery1.sql -o C:\Filer\SQLRespons.txt

   I staden for Datamaskinnamn kan du bruke IP-adresse (f.eks. 127.0.0.1 for din eigen maskin).

6. Køyr denne batchfila i Cmd-vinduet. Studer responsen ved å opne fila SQLRespons.txt.
7. Bruk IP-adressa til din maskin og til nokre av nabomaskinane (dei virtulelle) og du vil sjå at du også får tilgang til dei andre sine databaser dersom du kjenner brukarnamn og passord.

Lage og køyre SQL-skript i Visual C++

8. Lag ein Visual C++ applikasjon for innlesing av postnr og poststad :



9. Tilordn variablar m_postnr og m_poststad til edit-boksane og tilordn ein funksjon onleggin() til knappen .
10. Skriv inn denne koden (det som står i kursiv):

```
    void CFubasemedskriptDlg::Onlegginn()
{
    // TODO: Add your control notification handler code here
 UpdateData(TRUE);
 ofstream Utfil;
 Utfil.open("SQLQuery1.sql");
 Utfil<<"USE Fubase"<<endl;
 Utfil<<"INSERT INTO sted (postnr,poststed)
VALUES("<<m_postnr<<",'"<<m_poststad<<"')"<<endl;
 Utfil.close();
 system("DATABASE.BAT");
}
```

11. Skriv inn *#include <fstream.h>* under dei andre #include-setningane på toppen av fila. Velg menyvalg Build/Set Active Configuratin og sett den til Release slik at vi kan flytte exe-fila der vi vil etter at den er laga (f.eks. til katalog C:\Filer).
12. Ta Compile og Build og kopier exe-fila frå underkatalog Release til katalogen C:\Filer, og køyr den der.
13. Legg inn nokre poststader, og gå og sjå i databasen om dei er komne inn. Studer eventuelt fila SQLRespons.txt for å sjå etter feilmeldingar eller fila SQLQuery1.sql for å sjå om du har skrive rett.

Datamodellering 2017.

Øvelse med TCP-kommunikasjon programmert i C++

Desse øvingane køyrer vi på Windows 7 maskiner på Datalab.

1.

Lag ein katalog **Filer** på **D:\Cpp** (om du ikkje alt har det) .
Hent filene som ligg i zip-fila TCPiCpp2015.zip på
http://sksk.no/Tveita/Program/datasystemer.htm og ekstraher dei til Filer-katalogen din.

Opn katalogen TCPiCpp. Køyr programmet **main.exe** for å starte ein server. Bruk portnr. 5000. Server ligg no og lyttar etter kommunikasjon på port 5000.

Køyr programmet **client.exe**. Oppgi server si IP-adresse 127.0.0.1 (dvs din eigen maskin) Oppgi portadresse 5000. Følg med kva som skjer i begge vindu.

Svar n på spørsmålet om du vil sende ei fil.
Skriv inn ei melding på 1 ord (pga cin). Følg med kva som skjer i server-vinduet.
Svar n på spørsmål om du vil avslutte.
Hald fram med skriving av meldingar til du vil avslutte. Avslutt også server.

2.

Gå saman 2 og 2. Den eine skal køyre main.exe (server). Den andre client.exe.

Den som skal køyre server må finne ut IP-adressa si og gi til makker:
Gi kommando i CMD-vinduet: **ipconfig /all**, og noter deg IP-adressa di.

Den som skal køyre client, lagar ei lita tekstfil **fil.txt** og lagrar på **cpp-katalogen.**

Den som skal køyre server, startar **først** programmet **main.exe**. Bruk portnr. 6000

Den som skal køyre **client.exe**, oppgir makker si IP-adressa og portnr. 6000.
Svar j på spørsmålet om du vil sende fil, og oppgi filnamn med full bane
C:\Cpp\Filer….\fil.txt
Send diverse meldingar til makker.
Den som køyrer main, sjekkar at fila er komen fram.

3.
Lag chatte-program.

Start C++ i Visual Studio. Lag eit prosjekt for vanleg konsollprogram.

Opne programmet **main.cpp** og include-fila **wcomm.h** og fila **wcomm.cpp.**

Studer programmet **main.cpp** og include-fila **wcomm.h** og fila **wcomm.cpp**. Endre namn på main.cpp til **mainchatt.cpp** ved å velja **File/Save As.** Foreta endringar i **den** slik at server ikkje berre svarar OK men at du kan skrive inn fornuftige svar. Bruk cin.getline() for innput frå tastatur (men husk cin.ignore() første gangen. Sjå s. 191 i C++ boka)

Før linking (build) må denne setninga vera på plass f.eks. i wcomm.h:

```
#pragma comment(lib, "ws2_32.lib")
```

Starten på wcomm.h ser då slik ut:

```
#pragma comment(lib, "ws2_32.lib")
#include <winsock2.h>
#include <stdio.h>
#include <conio.h>
#include <iostream>
#include "fstream"
using namespace std;
```

Starten på mainchatt.cpp skal sjå slik ut (gjeld også client.cpp):

```
#include "wcomm.cpp"
#include <string>
using namespace std;
```

Lag exe-fil av mainchat, og kopier den frå Release-katalogen til Filer-katalogen.

Opne programmet **client.cpp** og erstatt cin>>melding med cin.getline(melding,50) så du også kan sende fleire ord (maks 50 teikn) frå klient-programmet. Legg også her inn cin.ignore() før while-sløyfa.

Prøv ut chatte-programma **mainchat** og **client** 2 og 2 i saman.

Måling med Labview

Når vi skal foreta målingar med Labview, skal måleresultata overførast frå Labview som køyrer på ein windows 7 maskin til eit Visual C++ program på den virtuelle maskinen vha TCP/IP – kommunikasjon med dei samme funksjonane som i **client**.

Labview køyrer eit serverprogram som sender data.

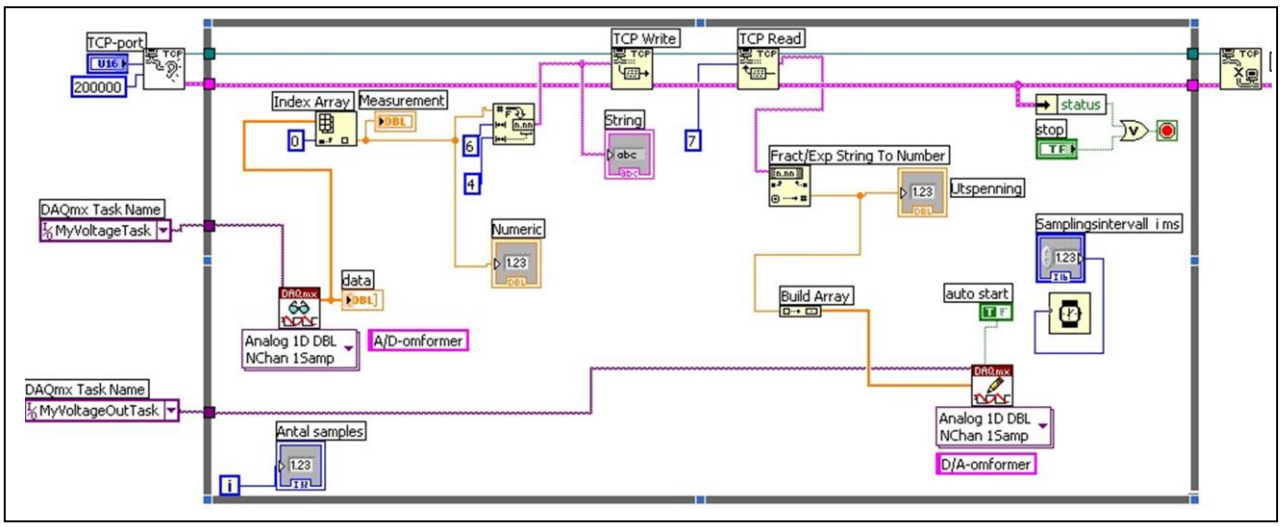
C++ - klientprogrammet **LoggingInnUt.cpp** har mellom anna ein funksjon **double Spenning()** for lesing av data som Labview sender.

C++ og Labview-sampling via TCP/IP.

Last ned katalogen http://sksk.no/Tveita/Program/TCPiCpp2015.zip både til virtuell maskin og til hovedmaskin (dersom du ikkje alt har gjort det). UnZip fila på begge maskiner. Her finn du dei filene du har bruk for når C++ skal utveksle data med Labview via TCP/IP.

Køyr Labview på Window7 maskin og C++ på den virtuelle maskinen. Opne Windows brannmur på begge maskiner. **Sjå til at "." er desimalteikn på Window7-maskin.**

Vi lar AD- og DA- omformar bli styrt av eit LabView Virtuelt Instrument **InnUtTCPNy.vi**:

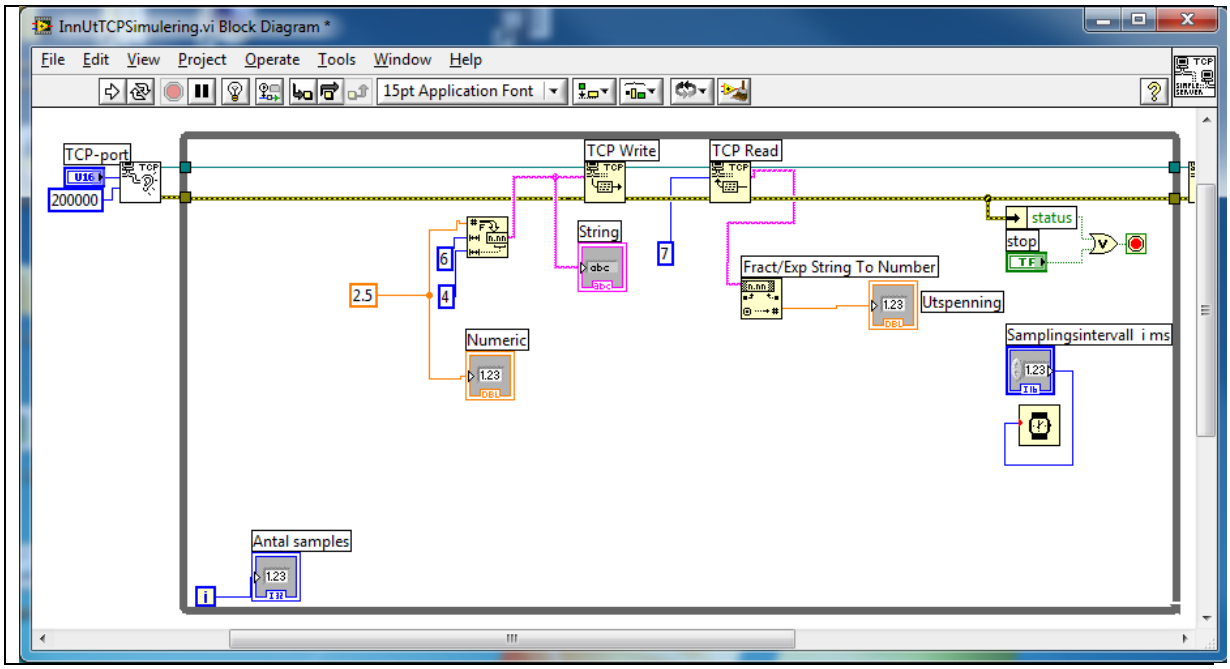


Denne VI-en startar ein TCP/IP-server som lyttar på ein valgt port. Når eit klientprogram tar kontakt på denne porten, startar ei while-sløyfe for lesing av ein spenningsverdi frå AD-omformar kanal 0 og sending av den til klienten som ein tekststreng av eit tal med lengde 6 teikn og med 4 desimalar. VI-en les også kvar gang ein spenningsverdi som kjem frå klienten som ein tekststreng med lengde 7, og legg den ut på DA-omformar 0.

C++ - klientprogrammet **LoggingInnUt.cpp** har ein funksjon **double Spenning()** for lesing av ein spenningsverdi frå AD-omformaren via nettet , og ein funksjon **void Utspenning(double U)** for å leggja ut ei spenning U til DA-omformaren via nettet. I dette testprogrammet vil spenninga U vera halvparten av den leste spenninga.

Framgangsmåte for testing av kommunikasjonen.

For å teste kommunikasjonen mellom Labview og den virtuelle maskinen, køyrer vi først ein forenkla versjon av Labview-programmet:



Denne simulerer at vi måler 2.5 V. Velg samplingsintervall 1000 ms og køyr denne med port 5000 (det er berre for den porten brannmuren er opna). Legg inn IP-adresse for Windows 7 maskinen og portnummer 5000 i programmet **LoggingInnUt.cpp** på den virtuelle maskinen, kompiler og køyr. Du skal då få fram målte spenningar på 2.5 V på skjermen kvart sekund. Du får også fram at programmet returnerer halve spenninga tilbake til Labview.

Framgangsmåte for testing av logginga.

På Datateknisk lab koplar utgangen frå frekvensgenerator til den «kvite boksen» sin differensielle A/D-inngang 0 på signalenhet og til oscilloskop-inngang 1. Kopl D/A - utgang 0 til oscilloskop-inngang 2 og Gnd til jord på oscilloskop. Juster offset og amplitude på frekvensgenerator slik at signalet ut er sinus-forma med amplitude ca 4 V.
Fjern DAQmx Task Name elementa, og bruk **Measurement I/O/ NIDAQmx/DAQ Assist** til leggja inn nye inn- og ut- kanalar.

Start LabView- programmet **InnUtTCPNy.vi** på Windowsmaskin**.** Bruk port 5000**. K**øyr **LoggingInnUt.cpp** på virtuell maskin. Sjå til at spenningane blir målt og at D/A-omformeren legg ut spenningsverdiar som er rekna ut (halve innlest verdi). LabView-programmet må alltid startast først.

Når du er ferdig med å teste ut LoggingInnUt.cpp, så bruk Save As for å lagre programmet som **Loggefunksjonar.cpp**. Komenter bort main(), og lagre på nytt. Desse funksjonane skal vi bruke i neste program.

Windowsprogram for logging med C++ og Labview via TCP/IP

I denne øvinga skal Labview-programmet køyrast på Windows7 – maskinen og loggeprogrammet LoggingWindows køyrast på den virtuelle maskinen.

1. Lage ein Windowsapplikasjon (MFC) som ser slik ut på den virtuelle maskinen:

Enkelt loggeprogram

Dersom du er i tvil om korleis du skal lage ein windowsapplikasjon med Timer og knappar, så gå tilbake til programmet Http- og ftp-funksjonar i C++ og sjå der.

2. Legg til variablar til Edit-boksane: m_Ts, m_U, og m_antal.
3. Gi startknappen namn IDC_Start og legg til funksjon OnStart med kode for å starte ein timer med 1000*m_Ts millisekund og nullstille m_antal.
4. Gi stoppknappen namn IDC_Stopp og legg til funksjon OnStopp med kode for å stanse timer. Gjer det samme i OnDestroy()-funksjonen.
5. Legg inn setninga *runclient("192.168.4.xx",5000);* i OnInitDialog() (IP-adressa til maskinen som køyrer Labview).
   Sett også *m_antal=0; m_Ts=1;UpdateData(FALSE);*
6. Legg til denne koden til OnTimer:
   m_antal++;
   m_U=Spenning();//Les ei spenning frå Labview sin server
   Utspenning(m_U/2);//Sender ei spenning til Labview-programmet
   UpdateData(FALSE);//Oppdaterer verdiane i skjemaet
7. Inkluder **Loggefunksjonar.cpp** i windowsapplikasjonen.
8. Sjå til at **wcomm.cpp**, **wcom.h** og **Loggefunksjonar.cpp** ligg i samme katalog som windowsapplikasjonen .
9. Start Labview-programmet **InnUtTCPNy.vi på fysisk maskin** og deretter **LoggingWindows** på virtuell maskin. Legg inn samplingsintervall (for eksempel 5 s) og start logging.

Datamodellering 2017.

Utvikling av internettbasert brukarapplikasjonsprogram med ASP-skript for database.

1. Vi må ha eit redigeringsprogram for enkle Web-sider. Vi kan bruke Notepad til å skrive html-kode eller Visual Studio til ein kombinasjon av html-kode og design.
2. Lag katalog C:\Inetpub\wwwroot\fubase. Last ned fila http://sksk.no/Tveita/Program/index.txt inn i eit dokument i Notepad og lagre det som (Alle filer, ikkje txt-fil) index.htm på katalog fubase.
3. Opne index.htm i Visual Studio, og du kjem inn i eit redigeringsprogram for html-sider.
4. Studer fila i Design-mode og i Source-mode. Vi har her eit Skjema (Form) med 2 Formfelt (Input Text og Input Button).
5. Under køyring av sida fyller du ut skjemaet, og ved klikk på **Send** skal ei anna side på web-serveren - SQLSIDE1.ASP - bli aktivert. På denne sida lagar vi eit skript som les kva du har svara i skjemaet og som så utfører ei sql-spørjing mot databasen vår. Lag førebels ei side SQLSIDE1.ASP med litt tekst på. Lagre begge sidene dine på C:\Inetpub\wwwroot\fubase.
6. Gå til **Server Manager/Dashboard/Configure this local server/Add roles and features**. Ekspander Web Server (IIS)/Webserver/Application Development/ og kryss av for ASP.
7. Finn IP-adressa di. Start Internet Explorer og skriv inn for eksempel: http://192.168.20.38/fubase/index.htm og du ser korleis sida di ser ut. Klikk på **Send**, og SQLSIDE1.ASP skal koma opp. Prøv også å få kontakt med naboen sin WEB-server.
8. **Dersom du ikkje alt har laga brukar testbruker1** med passord ***Sjekk1234*** i SQL-server, så gjer det her: Når webserveren (IIS) skal kommunisere med databaseserveren, må han logge seg inn på den. Vi etablerer difor ein ny Login: Start Microsoft SQL Server Management Studio. Ekspander Security/Logins. Studer kven som har innloggingsrettar. Høgreklikk på Logins og velg New Login. Velg Login Name: testbruker1. Velg SQL Server authentication og Password: Sjekk1234. Fjern avkryssing i Enforce password expiration. Velg Default Database: Fubase. Klikk på User Mapping og kryss av for Fubase, db_datawriter, db_datareader, db_owner og public. Klikk OK. Fjern primærnøkkel på tabell **Sted**.

9. Les kapittel 10.1-7 i læreboka og sjå figur 10.1 og 10.2. Webserveren (IIS) må ha ein mellomprogramvare (API) for å snakke med databasen. Vi brukar ODBC. Vi må då opprette eit Data Source Name (DSN). Gå til Administrative Tools/ODBC Data Sources (64-bits). Remove evt. Fubase, dersom den ligg der frå før.
10. Velg System DSN og ADD. Du får opp ei liste over drivarar, velg SQL Server og klikk Finish. Skriv inn Data Source Name: Fubase (valgfritt namn, men bruk Fubase her). Skriv inn beskrivelse: "Min lokale SQL Server". Velg Server: Du får opp ei liste der du vel XXXXX\DB der XXXXX er din maskin. Klikk Next.
11. Velg SQL Server authentication , velg Login ID: testbruker1 og Password: Sjekk1234. Velg Next og Default Database: Fubase. Du kan så velje Test Connection og får kanskje suksess.
12. Du skal no lage SQLSIDE1.ASP i f.eks. Visual Studio. Eit framlegg til side ligg på http://sksk.no/Tveita/SQLSIDE1.txt . Bruk samme framgangsmåte som i punkt 2 og 3

til å lagre innhaldet som SQLSIDE1.ASP i samme katalog som søkesida du laga i punkt 5.

13. Køyr søkesida og test ut. Studer HTML-koden og Javascript-koden på sida. Legg spesielt merke til at vi må endre Cursortype og Locktype når vi skal oppdatere data.
14. Legg inn andre SQL-skript i koden på SQLSIDE1.ASP og endre søkesida. Lag søking som involverer fleire tabellar. Test ut.

Lange SQL-skript kan du skrive i små bitar og addere dei f.eks. slik (husk mellomrom etter kvar del):

```
sqlstreng=”SELECT * FROM Sted   ”;
sqlstreng=sqlstreng+”WHERE postnr>’6000’ AND   ”;
sqlstreng=sqlstreng+”postnr<’8000’   ”;
RS.Open(sqlstreng);
```

Fasit for ein del oppgåver ligg på http://sksk.no/Tveita/Program/Fubasefiler.zip

Datamodellering 2017.

Å lage eit Visual C++ -program for databasetilgang.

Vi skal lage eit Visual C++ program der vi kan bla oss gjennom innhaldet i database-tabell, slette datapostar og leggja til nye datapostar.

Vi har alt laga ein database **Fubase** med ein tabell **Sted.** Gå inn i databasen, høgreklikk på tabellen Sted og velg **Design**. Fjern eventuell primærnøkkel frå kolonne **postnr** og endre namn på kolonna til **nr**. (postnr og poststed er så like namn at C++ kan bli forvirra)Vi må også laga ODBC-forbindelse til databasen med brukarnamn **Testbruker1** og passord **Sjekk1234,** men denne gangen må vi bruke **32-bits ODBC**. Bruk Fubase som ODBC-namn.

1. Start Visual C++ og velg **File/New/Project/MFCAppWizard(exe)** og velg Project name **Fubase** og lagringsplass på C:\Filer\Cpp. Klikk **OK**.
2. Velg så **Single document,** fjern avkryssing **Use Unicode lib.** Klikk **Next** 3 ganger**.**
3. Velg så **Database view without file support** og **ODBC** og **Dynaset** og klikk **DataSource**
4. Velg **Machine Data Source** og **Fubase**. **OK**.
5. Oppgi passord **Sjekk1234** og velg tabell **dbo.sted** og klikk **OK**.
6. Klikk **Finish** og så **OK** på neste dialog.
7. Du får opp eit skjema (ei **Form**) der du skal plassere boksar og knappar. (Kanskje må du opne **Dialog** på Resource View for å sjå Formen.) Foreta **Bulid** og **Execute** for å sjå om det er feil så langt. Dersom det kjem feilmelding «connection string may contain a password…», så dobbelklikk på feilmeldinga og komenter bort den setninga i programkoden som gir feilmelding. Er det feil må du kanskje starte på nytt.
8. I Class View: Velg klassen CFubaseView og i vinduet under får du fram alle funksjonar og variablar i klassen. I denne klassen er det definert ein variabel **m_pSet** som er ein peikar til eit **recordset** som vi hentar frå databasen når vi startar programmet. Dersom du dobbelklikkar på **m_pSet** kjem du inn i koden som viser at **m_pSet** er eit objekt av klassen **CFubaseSet**.
9. Velg også klassen CFubaseSet. Der ser du at denne klassen inneheld to variablar **m_nr** og **m_poststed**. Det fulle namnet til desse variablane blir **m_pSet->m_nr** og **m_pSet->m_poststed**. Dobbelklikkar du på dei, ser du at dei er av typen **int** eller l**ong** eller **CString** (sjå kommentar om CStringA).
10. No skal vi lage to Edit-boksar og knytte dei to variablane til Edit-boksane. Lag til skjemaet slik det er vist på dett biletet.

Sett namn IDC_Nr og IDC_Poststed på Edid-boksane og legg til variablar m_nr og m_poststed (hhv int og Cstring).

Foreta Build og Execute for å teste at programmet fungerer så langt.

11. Opne fila **FubaseView.cpp** og gå til funksjonen **CFubaseWiew**::**DoDataExchange()** og endre argumenta i funksjonane **DDX_Text()** frå **m_nr** til **m_pSet->m_nr** og frå **m_poststed** til **m_pSet->m_poststed.**
12. Velg **Build** og **Execute** og du ser du kan bla deg gjennom databasetabellen.
13. Vi skal no leggja funksjonalitet på knappane Endre, Legg til og Slett. Høgreklikk på knappane og velg Properties og gi dei ID: **IDC_Endre**, **IDC_Leggtil** og **IDC_Slett**. Opne ClasWizard og velg for kvar av knappane **BN_CLICKED/AddHandler/Edit Code**.
14. Legg inn denne koden(du kan klippe og lime frå dette dokumentet):

```
void CFubaseView::OnClickedLeggtil()
{
	// TODO: Add your control notification handler code here
m_pSet->AddNew();
//No kan du skrive inn nye verdiar i skjemafelta
//eller leggja inn nye verdiar f. eks. slik:
//m_pSet->m_nr="5000";
//m_pSet->m_poststad="Bergen";
UpdateData(TRUE);//Verdiane i skjemafeltet blir overført til
variablane
m_pSet->Update();//Databasen blir oppdatert med recordsettet
m_pSet->Requery();//Verdiane overført frå databasen til recordsettet
UpdateData(FALSE);//Skjemafelta blir oppdatert med første record
}
void CFubaseView::OnClickedSlett()
{
	// TODO: Add your control notification handler code here
m_pSet->Delete();//Slettar gjeldande post og oppdaterer database
m_pSet->Requery();
UpdateData(FALSE);
}
void CFubaseView::OnClickedEndre()
{
	// TODO: Add your control notification handler code here
	m_pSet->Edit();
//No kan du skrive inn nye verdiar i skjemafelta
//eller leggja inn nye verdiar f. eks. slik:
//m_pSet->m_nr="5000";
//m_pSet->m_poststad="Bergen";
UpdateData(TRUE);//Verdiane i skjemafeltet blir overført til
variablane
m_pSet->Update();//Databasen blir oppdatert med recordsettet
m_pSet->Requery();//Verdiane overført frå databasen til recordsettet
UpdateData(FALSE);//Skjemafelta blir oppdatert med første record
}
```

15. **Build** og **EXECUTE** og test ut.
16. For å sjekke ut kva andre kommandoar du kan bruke på eit recordsett, så skriv inn **m_pSet->** og det kjem opp ei liste med funksjonar.

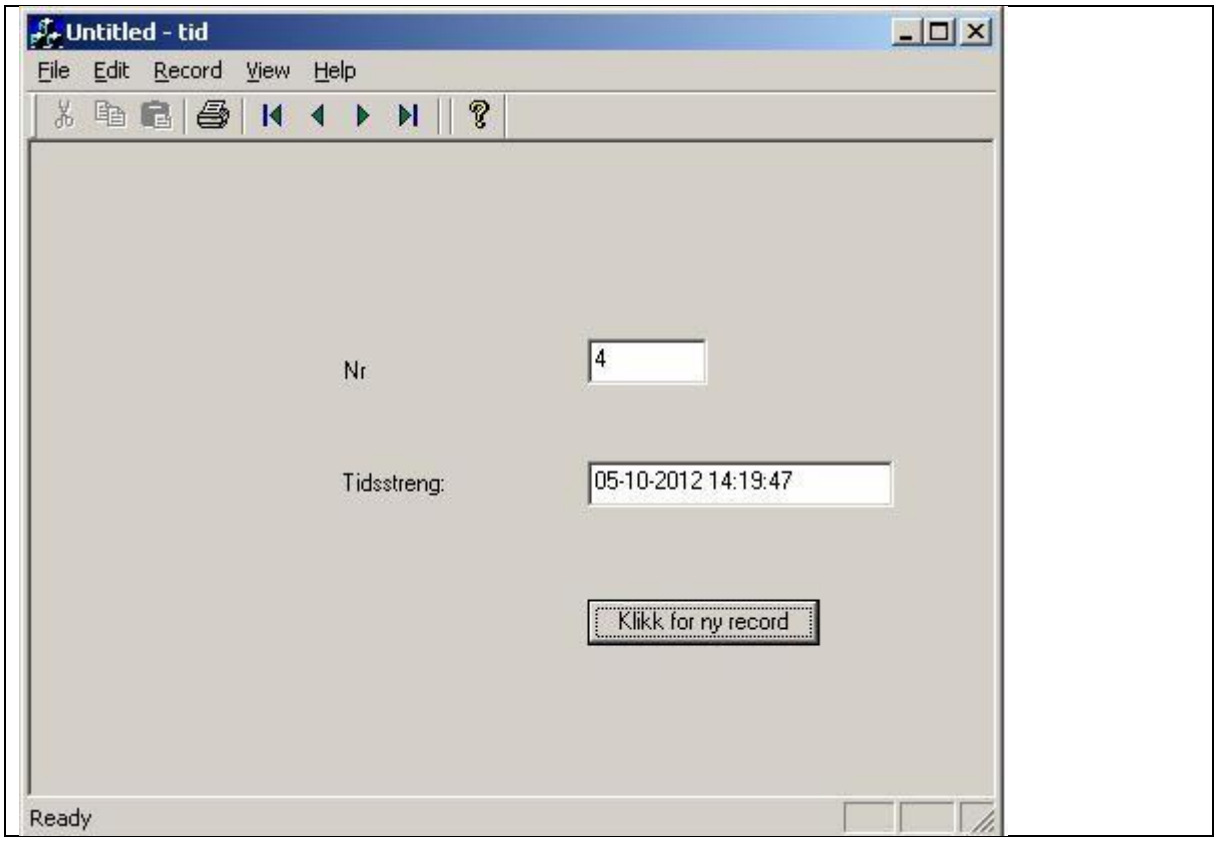
DateTime-oppgåve

Lag ein ny tabell i Fubase med kolonner Nr(int), Tidspunkt(DateTime) og Data(float).

Lag ein C++ - applikasjon som kvar gang du klikkar på knappen, legg inn ei ny linje i tabellen.
Nr aukar med 1 kvar gang
Tidspunkt blir lest av klokka i datamaskinen
Data er eit tilfeldig desimaltal mellom -5 og 5 generert av random-generator.

Lag ei web-side med asp-skript som skriv ut alle data i tabellen

C++-applikasjonen kan sjå noko slik ut (men bør innehalde visning av Data også):



DateTime i SQL-server og C++

Bruk mp_Set->m_tid som CTime. Du kan då putte ctid inn på recordsettet og den kjem som datetime i databasen.
For å visa tida i ein boks på skjermen, må den omformast til ein tekststreng av typen CString, her m_tidsstreng som er knytta til boksen på skjermen.

Her er tips til denne omforminga:

```
void CTidView::OnButton1()
{
	// TODO: Add your control notification handler code here
	antal++;
	time_t tid; //må inkludere <time.h>. Definerer struct tid
	time(&tid); //Hentar tid frå operativsystemet og legg inn i tid
	CTime ctid(tid);//Omformar til CTime-format
	CString stid=ctid.Format("%d-%m-%Y %H:%M:%S");//Lagar tekststreng
	m_pSet->AddNew();//Lagar ny rad i record-settet m_pSet
	m_pSet->m_tid=ctid;//Tid i tabellen må vera datetime som er kompatibel med CTime
	m_pSet->m_nr=antal;
	m_pSet->m_Data=Tilfeldigtal(); // Du må lage denne funksjonen
	m_tidsstreng=stid;    //Skal ikkje i databasen, berre til editboks
	UpdateData(FALSE);
	m_pSet->Update();//Oppdaterer databasen med dei nye verdiane
	m_pSet->Requery();//Hentar ut alle data frå databasen på nytt
	m_pSet->MoveLast();//Går til siste rad i record-settet
	m_tidsstreng=m_pSet->m_tid.Format("%d-%m-%Y %H:%M:%S");
	UpdateData(FALSE);//Oppdaterer verdiane i skjemaet
	}
```