



HØGSKOLEN  
I SØR-TRØNDELAG

# HØGSKOLEN I SØR-TRØNDELAG

## Avdeling for informatikk og e-læring - AITeL

<b>Kandidatnr:</b>	
<b>Eksamensdato:</b>	16.desember 2005
<b>Varighet:</b>	3 timer
<b>Fagnummer:</b>	LO116D
<b>Fagnavn:</b>	Programmering i Visual Basic
<b>Klasse(r):</b>	FU      1BABED
<b>Studiepoeng:</b>	6
<b>Faglærer:</b>	Svend Andreas Horgen tlf: 73 55 92 69 Grethe Sandstrak tlf: 73 55 95 61
<b>Kursassistent:</b>	Siri Wæhre Lien tlf: 73 55 91 54
<b>Hjelpemidler:</b>	Alle <b>skriftlige</b> hjelpemidler
<b>Oppgavesettet består av:</b>	4 oppgaver og totalt 5 sider (inkludert forsiden).
<b>Viktig:</b> Planlegg tiden godt. Dersom noe virker uklart i oppgavene, må du selv gjøre de nødvendige antakelser og få med disse i besvarelsen.	
<b>Lykke til!</b>	

## Oppgave 1 – Litt teori (10%)

Dersom du trenger det kan du relatere svaret ditt på denne oppgaven til følgende kodesnutt:

```
Private Sub Button1_Click(ByVal sender As System.Object,
                          ByVal e As System.EventArgs)
    Handles Button1.Click

    Dim mittTall, etAnnetTall As Double
    Static a As Integer
    mittTall = txtTall.Text
    etAnnetTall = Rnd()
    a = a + 1

End Sub
```

Så til selve oppgaven: Hva menes med følgende begrep? Forklar punktvis og med egne ord, helst så kort som mulig.

- datatype
- implisitt konvertering
- egenskap
- static
- funksjon

## Oppgave 2 – Variabler og kontrollstrukturer (30%)

Pizza-sjapper finner du stadig flere av, og i denne oppgaven skal du programmere noe av funksjonaliteten for en bestillings-applikasjon for *Pizza eXPressen AS*.

Programmet lar brukeren velge type pizza, størrelse og antall. Basert på disse opplysningene blir så pris beregnet og bestillingen presentert.

Kunden kan velge mellom tre ulike typer pizza – kun ett av tre valg er mulig: **Skinke**, **Pepperoni** og **Egenkomponert**. Ved valg av egenkomponert skal en inputboks gi brukeren mulighet til å skrive inn navn på ingrediens. Det skal være mulig å oppgi inntil 5 ingredienser, men det skal også være mulig å skrive teksten *ferdig* for å stoppe innlesningen.

Pris for pizzaene:

- Skinke: 75,-
- Pepperoni: 90,-
- Egenkomponert: 40,- pr. ingrediens

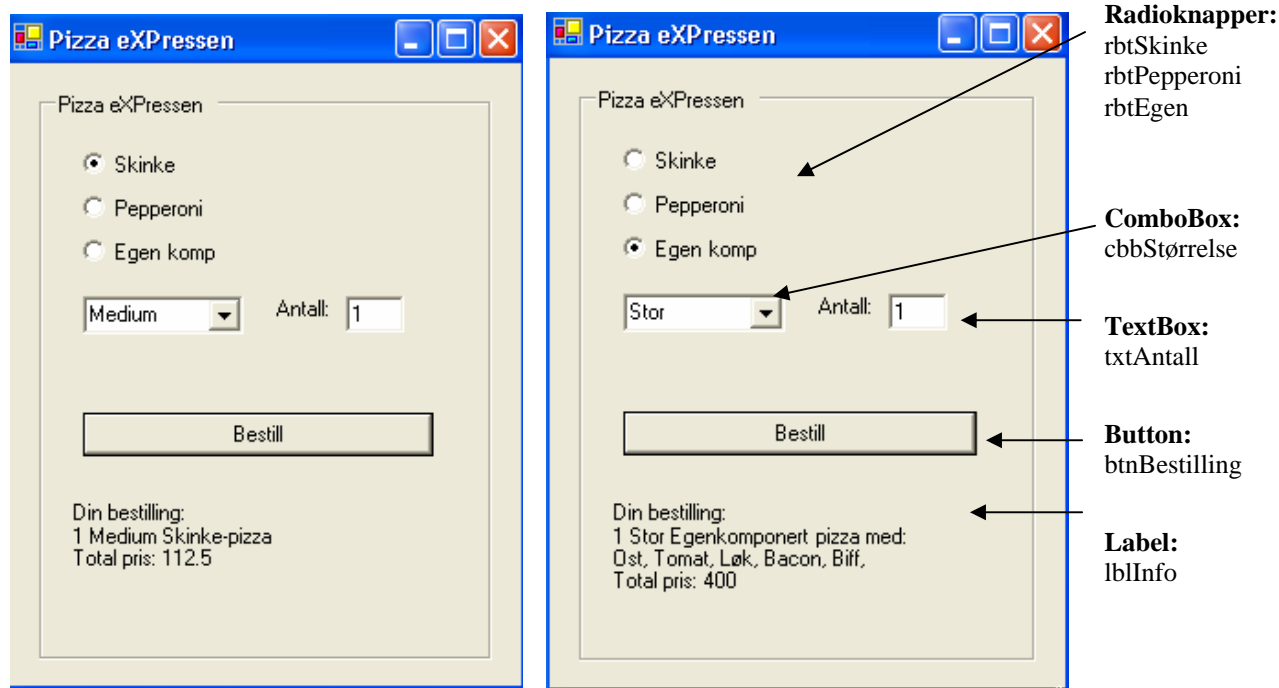
I tillegg skal det tas høyde for pizzaens størrelse ut fra følgende faktorer:

- Liten faktor = 1
- Medium faktor = 1,5
- Stor faktor = 2

Eksempel på utregning:

- 1 Liten Skinke-pizza vil koste:  $75 * 1 = 75$  kr
- 1 Medium Pepperoni-pizza vil koste:  $90 * 1,5 = 135$  kr.
- 1 Stor egenkomponert pizza med 3 ingredienser vil koste:  $40 * 3 * 2 = 240$  kr.

Selve oppgaven: Lag koden som må ligge bak knappen *btnBestilling* (Bestilling). Du kan gå ut ifra at brukeren oppfører seg eksemplarisk og taster inn forventede verdier. Navn på komponentene ser du av Figur 1.



Figur 1: Eksempel på bestilling av medium skinke-pizza til venstre og stor egenkomponert pizza til høyre.

## Oppgave 3 – Matriser og funksjoner (40%)

Denne problemstillingen fortsetter i oppgave 4 (filbehandling), men vi starter nå med matriser og funksjoner. Du kan gjøre oppgave 4 uten å få til denne, men må i så fall sette deg inn i problemstillingen først.

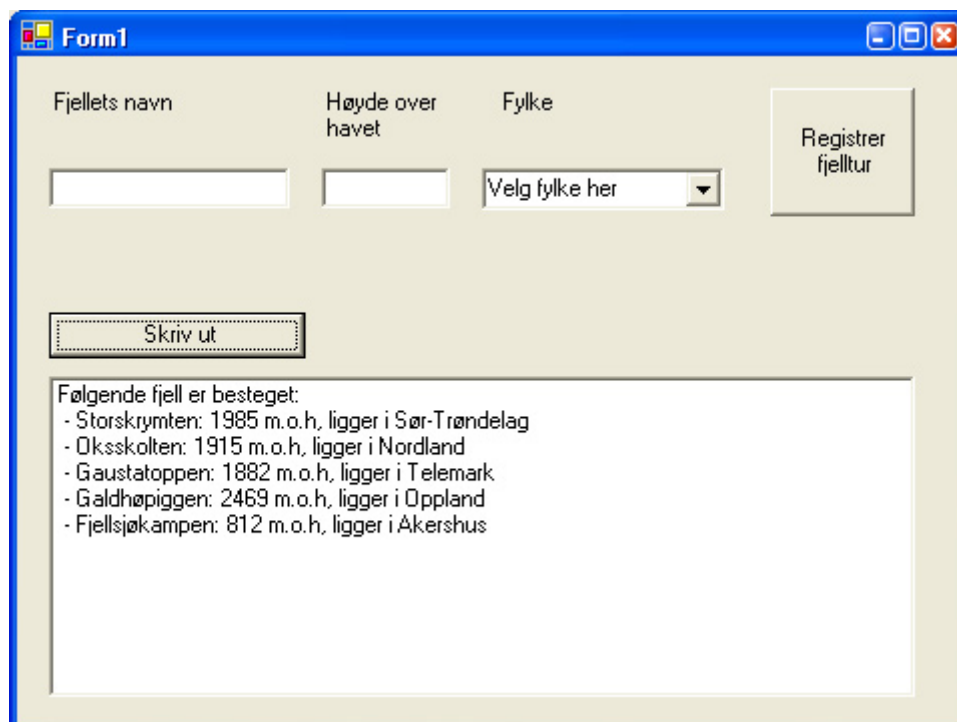
Det er mange fjell i Norge. Kari lurer på hvor mange hun egentlig har besøkt som er over 1800 meter. Hun vil derfor lage et program som gjør det mulig å registrere og hente fram informasjon om fjellturene i ettertid.

Kari vil registrere informasjon om navn, høyde (antall hele meter over havet) og fylke for hvert fjell hun har besteget. Hun velger å bruke *parallelle matriser* i første omgang (filer brukes først i oppgave 4). Det er lurt å lese alle deloppgaver før du starter med besvarelsen din.

- a. (5%) Følg instruksjonene nøye i denne oppgaven. Skriv fire setninger. En setning skal opprette en matrise med plass til nøyaktig 3 fjell. Matrisen skal hete `fjellTest`, og deretter skal du hardkode navnet på følgende tre fjell i matrisen: Galdhøpiggen, Oksskolten, Fjellsjøkampen.

Oppgave a er ment som hjelp for å komme i gang med resten av oppgaven, og matrisen `fjellTest` skal **ikke** brukes videre.

- b. (10%) Skriv den nødvendige koden som må til for å registrere informasjon om fjell: *navn*, *høyde* og *hjemfylke*. Velg selv om du vil lese inn informasjonen fra tekstfelt eller inputbokser, og husk å opprette matriser og variabler. Du kan se Figur 2 for et eksempel på hvordan grensesnittet kan se ut, men du trenger ikke bruke tid på å lage grensesnittet. Anta at det fins fra før.
- c. (10%) Lag kode som skriver ut informasjon om de registrerte fjellene i en listeboks, som vist i Figur 2:



Figur 2: Visning av alle fjell som er registrert i systemet.

- d. (5%) Lag en **funksjon** som heter `høydeStorNok` og som mottar to tall, en høyde og en terskel. Funksjonen skal sjekke om det første tallet (høyden) er større enn det andre (terskelen), og returnere `true` dersom det første er størst (eller evt. like stort).
- e. (10%) Bruk funksjonen i oppgave d i denne oppgaven. Skriv ut i listeboksen bare de fjell som er høyere enn et visst antall meter (brukeren får velge denne terskelen selv). Det holder å skrive ut bare fjellnavnet.

## Oppgave 4 – Filbehandling (20%)

Du kan gjøre oppgave 4 uten å få til oppgave 3, men må i så fall sette deg inn i problemstillingen først (se oppgave 3).

I oppgave 3 brukte Kari matriser for å lagre informasjon om fjell. Det går bra så lenge programmet kjører, men dataene går tapt i det programmet avsluttes. Det er altså behov for å skrive informasjonen fra matrisene til en fil eller en database. I denne oppgaven skal du bruke **filbehandling**.

- a. (8%) For langtidslagring på fil har du hovedsakelig to måter å lagre på: Sekvensielle og randomiserte filer. Du skal i denne oppgaven vise ( gjerne flere) mulige måter å organisere informasjonen på. Skisser og forklar kort. Gi også eksempel på hvilken kode som må til for å **opprette** de nødvendige filer med hver teknikk.
- b. (12%) Anta at det fins kode for å lagre informasjonen fra matrisen på en randomisert fil. Du skal nå lage kode som leser innholdet fra fil og legger inn i matrisestrukturere.