Høgskulen på Vestlandet

EKSAMENSOPPGAVER

Emnekode: **ELE205**

Emnenavn: **Videregående programmering**

Dato: **12.12.2022**

Eksamenstid:	4 timer
Antall oppgaver:	4
Antall sider (inkludert denne):	7
Antall vedlegg:	0 (ingen)
Tillatte hjelpemidler:	Håndskrevne notater, elektronisk kursmateriale og all tilgjengelig støttelitteratur (som ikke krever nettverkskommunikasjon). Samarbeid med, eller hjelp fra andre personer er ikke tillatt under eksamen.
Emneansvarlig:	Marcin Fojcik

Telefonnumrene lærere kan nås på under eksamen:

Marcin Fojcik (Førde):	917 90 947
Adis Hodzic (Bergen):	924 29 498
Olav Skjølingstad (Haugesund):	909 81 044
Frikk Hosøy Fossdal (Bergen):	938 29 805
Marcus Landschulze (Bergen):	400 11 133

Bokmål

Oppgave 1 (40%)

Her velger studentene ENTEN alternativ A ELLER alternativ B

Alternativ A

Anbefales valgt av studenter som har fulgt undervisning i Bergen eller Haugesund

Kilde for fig.: https://www.miljodirektoratet.no/globalassets/publikasjoner/klif2/publikasjoner/2828/ta2828.pdf

a) Du skal lage en klasse som heter OljeFelt.
Opprett følgende datamedlemmer: *feltNavn*, *eier*, *gjenværendeOljeMengde* (heltall), *land* (skal kunne ha verdiene Norge, Danmark og Storbritannia).

b) Opprett tilgangsmedlemmer for datamedlemmene.
Pass på at *feltNavn* ikke tillates satt til blank. Det skal gis feilmelding dersom dette forsøkes. Dersom *gjenværendeOljeMengde* forsøkes satt til negativt tall skal i stedet 0 settes inn.

c) Opprett et datamedlem ved navn *oljepris* (double). Dette datamedlemmet skal ikke tilhøre noe spesifikt objekt, men være felles for klassen.

d) Opprett et funksjonsmedlem ved navn *ØkonomiskFeltVerdi* som beregner økonomisk totalverdi utfra multiplikasjon mellom *oljepris* og *gjenværendeOljeMengde* og returner verdien.

e) Opprett et funksjonsmedlem ved navn *ToString*. Dette funksjonsmedlemmet skal overstyre (override) arvet *ToString*. *ToString* skal levere en tekststreng med objektets *feltNavn*, *eier* og *gjenværendeOljeMengde*, med passende informativ ledetekst.
(Måleenhet på *gjenværendeOljeMengde* er millioner fat.)

f) Opprett et funksjonsmedlem ved navn *SammeEierSom*, som tar et tilsvarende objekt som argument og returnerer true dersom argument-objektet har samme *eier*

som det objektet funksjonsmedlemmet kalles på (er 'kvalifisert med'). I motsatt fall skal false returneres.

g) Opprett et funksjonsmedlem ved navn *SammeEier*, som tar to oljefelt-objekter som argument og returnerer true dersom argument-objektene har samme eier. I motsatt fall skal false returneres. Dette funksjonsmedlemmet skal ikke tilhøre noe spesifikt objekt, men være felles for klassen.

h) Lag to konstruktører:
- En konstruktør som tar argumentverdi for *feltNavn, eier, gjenværendeOljeMengde*, *land*.
- En konstruktør som bare tar ett argument: *feltNavn.*

i) Vi har gitt et interface på formen

```
interface IProduksjonsFelt
{
        public void SkrivUtInfo();
        public string FeltNavn
        {
            get; set;
        }
}
```

Du skal la klassen OljeFelt oppfylle ('implementere') interfacet. *FeltNavn* skal allerede være tatt hånd om i oppgave b), mens *SkrivUtInfo* må implementeres. Denne metoden (funksjonsmedlemmet) skal skrive ut verdien av alle objektets datamedlemmer. Utskriften skal være slik at det kommer ett datamedlem per linje.

j) Opprett operatoren == som skal gi true dersom de to objektene den brukes på har samme *feltNavn*.

k) Lag et testprogram som oppretter tre objekter av typen Oljefelt. Legg de tre objektene inn i en liste av oljefelt. Alle data du legger inn velger du fritt. Men bruk gjerne navnene *Ekofisk, Troll og Statfjord*. Sorter lista etter stigende lengde på *feltNavn*. Utskrift er ikke krevd i denne deloppgaven.

l) Opprett fem objekter av typen OljeFelt (eller bruk de tre fra forrige deloppgave og opprett i tillegg to til). Legg de fem objektene inn i en liste av IProduksjonsFelt. Alle data du legger inn velger du fritt. Men bruk gjerne navnene *Ekofisk, Gullfaks, Oseberg*, *Troll og Statfjord*.
Kjør en løkke som utfører *SkrivUtInfo* på alle objektene i lista.

Du skal levere følgende: hele prosjektmappen (som én zip-fil) som heter Oppgave1_A.

Oppgave 1 (40%)

Her velger studentene ENTEN alternativ A ELLER alternativ B

Alternativ B

Anbefales valgt av studenter som har fulgt undervisning i Førde

En bedrift produserer biler og sykler. Bilene har pris (*tilfeldig heltall mellom 100 og 10000*), bensinmotor (*motorstørrelsen skal være random heltall fra 1 til og med 6 liter*) og kan ha oppvarming av seter. Syklene har pris og kan ha elektrisk motor i tillegg. Alle kan se alle priser, men ingen kan endre priser.
Skriv et C#-program som:

a) Lager datasamling med minst 4 bil-objekter og minst 4 sykkel-objekter.

b) Finner sum av pris av alle produkter.

c) Sorterer og skriv ut alle elementer (biler og sykler) fra størst til minst pris.

d) Finner hvor mange ulike bilmotorer som finnes i samlingen.

e) Finner hva som koster mest: alle elektriske sykler eller alle ikke-elektriske sykler.

Programmet skal ha god struktur, være brukervennlig og må behandle ugyldige eller ikke-korrekte data.

Du skal levere følgende: hele prosjektmappen (som én zip fil) som heter Oppgave1_B.

Vurdering for alternativ 1B:
35% for algoritme og strukturen
30% for beregning
15% for utskrift
10% for brukervennlighet
10% for kommentar og forklaring

Oppgave 2 (35%)

a) Lag et C#-program for idrettslag. Programmet skal inneholde informasjon om idrettsutøvere. Hver idrettsutøver skal ha navn, idrett (Enten Løp100Meter, HøydeHopp eller Svømming100Meter) og personlig rekord (i sin idrett). Personlig rekord skal være i sekunder eller meter (meter er for HøydeHopp).
Opprett en liste med minst 10 idrettsutøvere. Alle tre idretter nevnt ovenfor skal være representert i lista. Du kan opprette disse 10 objektene direkte i koden, altså uten å lese data fra tastatur. Merk at en idrettsutøver kan være registrert på mer enn en enkelt idrett.

b) I hovedprogrammet, legg inn kode som gjør følgende:
   - Finn og skriv ut den person som løper raskest.
   - Finn og skriv ut den person som har svakest personlig rekord i svømming.

   Ved utskrift skal tider vises med to desimaler.

c) Finn og skriv ut gjennomsnittlig hoppehøyde, med to desimaler.

d) Sjekk om det finnes personer som er registrert på flere idretter. Du trenger ikke skrive ut hvilke personer det eventuelt gjelder, bare om det finnes slike tilfeller eller ei.

Programmet skal ha god struktur, være brukervennlig og være robust mot ugyldig input fra brukeren.

Du skal levere følgende: hele prosjektmappen (som én zip fil) som heter Oppgave2.

Oppgave 3 (10%)

Implementer en rekursiv metode som beregner x^n, der x er et desimaltall og n er et heltall. Husk at, per definisjon, $x^0 == 1$. Vis bruk av metoden minst en gang i et program.

Du skal levere følgende: hele prosjektmappen (som én zip fil) som heter Oppgave3.

Oppgave 4 (15%)

a) **1%** Hva betyr **private** modifikator?

b) **2%** Forklar hva en **rekursiv** metode er.

c) **2%** Hva betyr det at en datatype er **generisk**?

d) **1%** Nevn minst en Collection som ikke tillater duplikater av verdier?

e) **3%** Navngi tre generiske collections vi har snakket om og beskriv for hver av dem kort et eksempel (problem) der det er passende å bruke den aktuelle datatypen. Svaret kan gis som tekst. Det er ikke nødvendig med kode, men kode er lov dersom du ønsker det.

f) **2%** Hva er grunnen til at vi har brukt interface IComparable i emnet?

g) **4%** (Merk: du skal bare gjøre en av deloppgavene g) og h). Du vil ikke få poeng for begge) Deloppgave g): Vi har to klasser:

```
class Test1
{
   static Test1()
   {
      Console.WriteLine("TEST1");
   }
}
```

og

```
class Test2
{
   public Test2()
   {
      Console.WriteLine("TEST2");
   }
}
```

Hva skal vi se på skjermen etter at koden nedenfor er kjørt, og **hvorfor**?

```
      ...
Test1 t11 = new Test1();
Test1 t12 = new Test1();
Test2 t21 = new Test2();
Test2 t22 = new Test2();
      ...
```

h) **4%** (Merk: du skal bare gjøre en av deloppgavene g) og h). Du vil ikke få poeng for begge) Deloppgave h): Hva er orden til følgende programkode:

```
void hemmeligF2(int tab[], int n)
{
   for(int i = 0; i < n; ++i)
   {
      for(int j = 0; j < n; ++j)
      {
         tab[j]++;
      }
   }
   for(int k = 0; k < n; ++k)
   {
      tab[k]+= 5;
   }
}
```

Du skal levere følgende: svarer på spørsmål som tekst dokument (pakket som én zip fil) som heter Oppgave4

Høgskulen på Vestlandet

EKSAMENSOPPGÅVER

Emnekode: **ELE205**

Emnenamn: **Vidaregåande programmering**

Dato: **12.12.2022**

Eksamenstid:	4 timar
Antal oppgåver:	4
Antal sider (inkludert denne):	7
Antal vedlegg:	0 (ingen)
Tillatne hjelpemiddel:	Handskrivne notat, elektronisk kursmateriale og all tilgjengeleg støttelitteratur(som ikkje krev nettverkskommunikasjon) Samarbeid med, eller hjelp frå andre personar er ikkje tillate under eksamen.
Emneansvarlig:	Marcin Fojcik

Telefonnummer som lærarane kan nås på under eksamen:

Marcin Fojcik (Førde):	917 90 947
Adis Hodzic (Bergen):	924 29 498
Olav Skjølingstad (Haugesund):	909 81 044
Frikk Hosøy Fossdal (Bergen):	938 29 805
Marcus Landschulze (Bergen):	400 11 133

Bokmål

Oppgave 1 (40%)

Her vel studentane ENTEN alternativ A ELLER alternativ B

Alternativ A

Vært anbefalt valt for studentar som har følgt undervisning i Bergen eller Haugesund

Kilde for fig.: https://www.miljodirektoratet.no/globalassets/publikasjoner/klif2/publikasjoner/2828/ta2828.pdf

a) Du skal lage ein klasse som heiter OljeFelt.
Opprett følgjande datamedlemmer: *feltNavn*, *eier*, *gjenværendeOljeMengde* (heltall), land (skal kunne ha verdiane Norge, Danmark og Storbritannia).

b) Opprett tilgangsmedlemmer for datamedlemmene.
Pass på at *feltNavn* ikkje er tillate å settast til blank. Det skal gis feilmelding dersom dette vært forsøkt. Dersom *gjenværendeOljeMengde* vært forsøkt sett til negativt tal skal i staden 0 settast inn.

c) Opprett ein datamedlem ved namn *oljepris* (double). Denne datamedlemmen skal ikkje tilhøyra noko spesifikt objekt, men være felles for klassen.

d) Opprett ein funksjonsmedlem ved namn *ØkonomiskFeltVerdi* som bereknar økonomisk totalverdi ut frå multiplikasjon mellom Oljepris og *gjenværendeOljeMengde* og returner verdien.

e) Opprett ein funksjonsmedlem ved namn *ToString*. Denne funksjonsmedlemmen skal overstyre (override) arva *ToString*. *ToString* skal levere ein string med objektet sitt *feltNavn*, *eier* og *gjenværendeOljeMengde*, med passande informativ leietekst. (Måleeining på *gjenværendeOljeMengde* er millionar fat.)

f) Opprett ein funksjonsmedlem ved namn *SammeEierSom*( ), som tek eit tilsvarende objekt som argument og returnerer true dersom argument-objektet har same *eier* som det objektet funksjonsmedlemmen vært kalla på (er 'kvalifisert med'). I

motsett fall skal false returnerast.

g) Opprett ein funksjonsmedlem ved namn *SammeEier*, som tar to oljefelt-objekt som argument og returnerer true dersom argument-objektene har same eier. I motsett fall skal false returnerast. Denne funksjonsmedlemmen skal ikkje tilhøyre noko spesifikt objekt, men være felles for klassen.

h) Lag to konstruktørar:
   - Ein konstruktør som tar argumentverdi for *feltNavn*, *eier*, *gjenværendeOljeMengde*, land.
   - Ein konstruktør som berre tar eit argument: *feltNavn*.

i) Vi har gitt eit interface på forma

```
interface IProduksjonsFelt
{
        public void SkrivUtInfo();
        public string FeltNavn
        {
            get; set;
        }
}
```

Du skal la klassen OljeFelt oppfylle ('implementere') interfacet. *FeltNavn* skal allereie være tatt hand om i oppgåve b), mens *SkrivUtInfo* må implementerast. Denne metoden (funksjonsmedlemmen) skal skrive ut verdien av alle objektet sine datamedlemmer. Utskrifta skal være slik at det kjem ein datamedlem per linje.

j) Opprett operatoren == som skal gi true dersom dei to objekta den vært brukt på har same *feltNavn*.

k) Lag et testprogram som opprettar tre objekt av typen Oljefelt. Legg dei tre objekta inn i ei liste av oljefelt. Alle data du legg inn vel du fritt. Men bruk gjerne namna *Ekofisk, Troll og Statfjord*. Sorter lista etter stigande lengde på *feltNavn*. Utskrift er ikkje kravd i denne deloppgåva.

l) Opprett fem objekt av typen OljeFelt (eller bruk dei tre frå førre deloppgåve og opprett i tillegg to til). Legg dei fem objekta inn i ei liste av IProduksjonsFelt. Alle data du legg inn vel du fritt. Men bruk gjerne namna *Ekofisk, Gullfaks, Oseberg*, *Troll og Statfjord*.
Køyr ei løkke som utfører *SkrivUtInfo* på alle objekta i lista.

Du skal levere følgande: heile prosjektmappa (som éi zip-fil) som heiter Oppgave1_A.

Oppgave 1 (40%)

Her vel studentane ENTEN alternativ A ELLER alternativ B

Alternativ B

Vært anbefalt valt for studentar som har følgt undervisning i Førde

Ei bedrift produserer bilar og syklar. Bilane har pris (*tilfeldig heiltal mellom 100 og 10000*), bensinmotor (*motorstørrelsen skal være random heiltal frå 1 til og med 6 liter*) og kan ha oppvarming av seter. Syklane har pris og kan ha elektrisk motor i tillegg. Alle kan sjå alle prisar, men ingen kan endre prisar.
Skriv et C#-program som:

a) Lagar datasamling med minst 4 bil-objekt og minst 4 sykkel-objekt.

b) Finn sum av pris av alle produkt.

c) Sorterer og skriv ut alle element (bilar og syklar) frå størst til minst pris.

d) Finn kor mange ulike bilmotorar som finnest i samlinga.

e) Finn kva som kostar mest: alle elektriske syklar eller alle ikkje-elektriske syklar.

Programmet skal ha god struktur, være brukarvennleg og må behandle ugyldige eller ikkje-korrekte data.

Du skal levere følgande: heile prosjektmappa (som éi zip-fil) som heiter Oppgave1_B.

Vurdering for alternativ 1B:
35% for algoritme og strukturen
30% for berekning
15% for utskrift
10% for brukarvennlegheit
10% for kommentar og forklaring

Oppgåve 2 (35%)

a) Lag eit C#-program for idrettslag. Programmet skal innehalde informasjon om idrettsutøvarar. Kvar idrettsutøvar skal ha namn, idrett (Enten Løp100Meter, HøydeHopp eller Svømming100Meter) og personleg rekord (i sin idrett). Personleg rekord skal være i sekund eller meter (meter er for HøydeHopp).
Opprett ei liste med minst 10 idrettsutøvarar. Alle tre idrettar nemnt ovanfor skal være representert i lista. Du kan opprette desse 10 objektene direkte i koden, altså utan å lese data frå tastatur. Merk at ein idrettsutøvar kan være registrert på meir enn ein enkelt idrett.

b) I hovudprogrammet, legg inn kode som gjer følgjande:
   - Finn og skriv ut den person som løp raskast.
   - Finn og skriv ut den person som har svakast personleg rekord i svømming.

   Ved utskrift skal tider visast med to desimalar.

c) Finn og skriv ut gjennomsnittleg hoppehøgde, med to desimalar.

d) Sjekk om det finnest personar som er registrert på fleire idrettar. Du treng ikkje skrive ut kva personar det eventuelt gjeld, berre om det finnest slike tilfelle eller ei.

Programmet skal ha god struktur, være brukarvennleg og være robust mot ugyldig input frå brukaren.

Du skal levere følgande: heile prosjektmappa (som éi zip-fil) som heiter Oppgave2.

Oppgåve 3 (10%)

Implementer ein rekursiv metode som bereknar x^n, der x er eit desimaltal og n er et heiltal. Husk at, per definisjon, $x^0 == 1$. Vis bruk av metoden minst ein gong i et program.

Du skal levere følgande: heile prosjektmappa (som éi zip-fil) som heiter Oppgave3.

Oppgåve 4 (15%)

a) **1%** Kva betyr **private** modifikator?

b) **2%** Forklar kva ein **rekursiv** metode er.

c) **2%** Kva betyr det at ein datatype er **generisk**?

d) **1%** Nemn minst ein Collection som ikkje tillèt duplikat av verdiar?

e) **3%** Namngi tre generiske collections vi har snakka om og beskriv for kvar av dei kort eit eksempel (problem) der det er passande å bruke den aktuelle datatypen. Svaret kan gis som tekst. Det er ikkje nødvendig med kode, men kode er lov dersom du ønskjer det.

f) **2%** kva er grunnen til at vi har brukt interface IComparable i emnet?

g) **4%** (Merk: du skal berre gjere ei av deloppgåvene g) og h). Du vil ikkje få poeng for begge) Deloppgåve g): Vi har to klassar:

```
class Test1
{
   static Test1()
   {
      Console.WriteLine("TEST1");
   }
}
```

og

```
class Test2
{
   public Test2()
   {
      Console.WriteLine("TEST2");
   }
}
```

Kva skal vi sjå på skjermen etter at koden nedanfor er køyrt, og **Korfor**?

```
      ...
Test1 t11 = new Test1();
Test1 t12 = new Test1();
Test2 t21 = new Test2();
Test2 t22 = new Test2();
      ...
```

h) **4%** (Merk: du skal berre gjere ei av deloppgåvene g) og h). Du vil ikkje få poeng for begge) Deloppgåve h): Kva er orden til følgjande programkode:

```
void hemmeligF2(int tab[], int n)
{
   for(int i = 0; i < n; ++i)
   {
      for(int j = 0; j < n; ++j)
      {
         tab[j]++;
      }
   }
   for(int k = 0; k < n; ++k)
   {
      tab[k]+= 5;
   }
}
```

Du skal levere følgande: heile prosjektmappa (som éi zip-fil) som heiter Oppgave 4