



**Høgskolen i Gjøvik**

**Avdeling for Teknologi**

---

# **E K S A M E N**

**FAGNAVN:** **Grunnleggende datakunnskap og programmering**

**FAGNUMMER:** **L 182 A**

**EKSAMENS DATO:** **14. desember 1999**

**KLASSE:** **99HINDA / 99HINDB / 99HINEA  
99HDMUA / 99HDMUB / 99HINGA**

**TID:** **09.00-13.00**

**FAGLÆRER:** **Frode Haug**

**ANTALL SIDER UTLEVERT:** **7 (inkludert denne forside)**

**TILLATTE HJELPEMIDLER:** **Kun læreboka "OOP in C++"**

- Kontroller at alle oppgavearkene er tilstede.
- Innføring med penn, evt. trykkblyant som gir gjennomslag.  
Pass på at du ikke skriver på mer enn ett innføringsark om gangen (da det blir uleselige gjennomslag om flere ark ligger oppå hverandre når du skriver).
- Ved innlevering skilles hvit og gul besvarelse og legges i hvert sitt omslag.  
Oppgavetekst, kladd og blå kopi beholder kandidaten.
- Ikke skriv noe av din besvarelse på oppgavearkene.
- Husk kandidatnummer på alle ark.

Dette eksamenssettet består av tre ulike oppgavetyper:

Oppgave 1 omhandler teori fra datateknikk-delen.

Oppgave 2 omhandler hva som blir utskriften fra noen ulike program.

Oppgave 3 omhandler et litt større programmerings-case.

**NB:** De tre oppgavene er totalt uavhengige og kan derfor løses separat.

## Oppgave 1 (20 %)

### a) Internet:

a1) Forklar kort (max. ½ A4-ark) de mest typiske funksjonene/mulighetene i en browser.

a2) Hvilke tre metoder for mottak, (videre)sending og oppbevaring av email har vi ?

### b) Datamaskinen (periferiutstyr / innmat):

b1) Redegjør kort (max. ½ A4-ark) omkring printere/skrivere.

b2) Redegjør kort (max. ½ A4-ark) omkring RAM og ROM.

b3) Redegjør kort (max. ½ A4-ark) omkring hva et operativsystem er.

b4) Redegjør kort (max. fire setninger pr. punkt) omkring:

b4-a) USB

b4-b) RISC

b4-c) Multitasking

### c) Datasikkerhet:

c1) Nevn fem ulike eksempler på sabotasje ifm. datakriminalitet.

c2) Nevn fem karakteristika på datavirus.

### d) Personvern / etikk:

d1) For hvem, innen hvilke områder og hvilken type registre gjelder personvernloven for ?

d2) Det blir lagret opplysninger om en person i et EDB-basert personregister.

Hvilke tre generelle rettighetsregler gjelder for vedkommende ?

d3) Nevn fem kategorier for etiske problemområder ifm. informasjonsteknologi.

## Oppgave 2 (20 %)

### a) Hva blir utskriften fra følgende program:

```
#include <iostream>

using namespace std;

int main() {
    char tekst[] = "RISKREMEN";
    int i = 4;

    while (tekst[i] != 'K') {
        cout << tekst[i] << '\n';

        i = ( i == 8) ? 0 : i+1;

        if (tekst[i] == 'E') {
            cout << "eeee!\n"; ++i;
        } else if (tekst[i] == 'R')
            i += 3;
    }
    return 0;
}
```

### b) Hva blir utskriften fra følgende program:

```
#include <iostream>
#include <cstring>

using namespace std;

void funk(char txt[], int n = 1, char ch = 'L')
{ txt[n] = ch; }

void funk(char txt[], char ch)
{ txt[2] = ch; ch = 'Å'; }

void funk(char txt[], char & ch, int n)
{ ch = txt[n]; strcpy(txt, "MOLBO"); }

int main() {
    char tekst[] = "FRODE";
    char tegn = 'Ø';

    funk(tekst);          cout << tekst << '\n';
    funk(tekst, tegn);    cout << tekst << " " << tegn << '\n';
    funk(tekst, 3, 'T');  cout << tekst << '\n';
    funk(tekst, tegn, 3); cout << tekst << " " << tegn << '\n';
    funk(tekst, 0, tegn); cout << tekst << '\n';
    return 0;
}
```

# Oppgave 3 (60 %)

**NB:** Les hele teksten for denne oppgaven nøye, før du begynner å besvare noe som helst.

Gjør dine egne forutsetninger dersom du finner oppgaveteksten upresis eller ufullstendig.

Gjør i så fall rede for disse forutsetningene først i besvarelsen din.

## Innledning

I denne oppgaven skal du lage et lite program som holder orden på utlån av traller i den oppdiktede Dyreparken på Gjøvik. Slike traller lånes ut til publikum (for en liten pengesum pr. dag), slik at de lettere kan frakte med seg den håndbagasjen (ekstra klær, mat, drikke, o.l) de har med seg til parken. Trallene lånes ut fra ei lita bu like ved parkens inngang. På en datamaskin inne i denne bua så kjører det programmet som du her skal lage. Trallene står ”linet opp” utenfor bua. På hver tralle er det påklistret et unikt nummer i intervallet 1-30. Når en besøkende ønsker å låne ei tralle, så tar vedkommende bare ei ledig ei utenfor bua, og går deretter inn i bua og oppgir sitt navn og trallenummer, og betaler. Under kjøring av programmet vil alle dataene ligge i datamaskinens primærhukommelse.

## Datastrukturen

I vedlegget kan du på den første siden se deklarasjoner/definisjoner av globale const'er, en struct, en array og en int. Dette skal være alt det du trenger av slike variable for å løse denne eksamensoppgaven. Datastrukturen vil derfor bestå av en array av struct'en 'tralle' og en global variabel som forteller totalt hvor mange av de 30 trallene som for tiden er utlånt. Tralle-arrayen er ikke sortert på noe kriterie, ei heller låner de besøkende trallene i stigende nummerrekkefølge. Dvs. den globale variabelen 'ant\_laant' vil ikke inneholde siste indeks som så langt er tatt i bruk av arrayen "status", men den vil inneholde hvor mange av de 30 "skuffene" i arrayen som totalt er i bruk i øyeblikket.

**Under løsningen av denne eksamensoppgaven skal du bruke denne datastrukturen !**

**Studer vedlegget nøye, som inneholder mange opplysninger om const'er, struct'en, globale variable, et hovedprogram og hjelpefunksjonen 'les\_nr' (som du også skal benytte deg av).**

## Oppgaven

### a) Lag funksjonen “void utlaan()” (Kommandoen ‘U’)

Dersom alle trallene for øyeblikket er utlånt, skal det komme en melding om dette. I motsatt fall så leses det inn et trallenummer i intervallet 1-30. Dersom denne trallen er utlånt, skal det også komme en melding. Ellers markeres den som utlånt og lånerens navn registreres i den aktuelle struct'en. Husk også på å oppdatere variabelen 'ant\_laant'. Datastrukturen skrives så til fil (ved å tilkalle funksjonen du lager i oppgave 3d).

### b) Lag funksjonen “void innlevering()” (Kommandoen ‘I’)

Dersom ingen traller er utlånt, så kommer en melding om dette. I motsatt fall så leses et lovlig trallenummer. Om trallen ikke er utlånt, så kommer det også en melding. Ellers markeres den som ledig, 'ant\_laant' oppdateres, lånerens navn skrives på skjermen og datastrukturen skrives til fil.

**c) Lag funksjonen “void les\_fra\_fil()”**

ASCII-filen ”TRALLER.DTA” inneholder opplysninger kun om de trallene som for øyeblikket er utlånt. På hver linje ligger det alle opplysninger om ett slikt lån. Dvs. på starten av hver linje ligger et trallenummer, deretter ett blankt tegn og til slutt lånerens full navn (ofte flere ord). Skriv funksjonen som leser alle linjene på filen. For hver linje så oppdateres aktuell struct i arrayen ’status’ med at trallen er utlånt og med lånerens navn. Husk også på å oppdatere variabelen ’ant\_laant’ korrekt, og å avbryte innlesningen om det skulle være mer enn 30 linjer på filen.

**NB:** Legg merke til at denne funksjonen tilkalles automatisk i starten av ’main’.

**d) Lag funksjonen “void skriv\_til\_fil()”**

Funksjonen går gjennom hele arrayen ’status’ og skriver ut opplysninger om de trallene som er utlånt til ASCII-filen ”TRALLER.DTA”. Datene er på samme format som angitt i oppgave 3c.

**e) Lag funksjonen “void finn\_person()” (Kommandoen ‘F’)**

Først spørres det om og leses inn et navn. Deretter blir det sjekket om denne personen har lånt noen tralle. Om så er tilfelle, så skrives trallens nummer til skjermen. I motsatt fall kommer også en melding.

**f) Lag funksjonen “void utlaan2()” (Kommandoen ‘U’ – ny)**

Her skal du lage en forbedret (?) versjon av utlåns-funksjonen. Denne skal fungere noe annerledes enn den laget i oppgave 3a. Nå skal ikke de besøkende lengre finne seg en egen tralle, og bare melde fra om nummeret i bua. Nå skal den første ledige med lavest nummer bli utlånt til vedkommende!

Funksjonen må derfor lete fra starten av arrayen ’status’ etter første ledige trallenummer. Om et slikt ikke finnes, så kommer en melding om det. Ellers så skrives den ledige trallens nummer ut på skjermen. Som i oppgave 3a oppdateres deretter variabelen ’ant\_laant’, trallen markeres som utlånt, lånerens navn registreres og datastrukturen skrives til fil.

**NB:** Oppgave 3f baserer seg altså på at trallene utlånes i stigende nummerrekkefølge. Men siden traller innleveres i tilfeldig rekkefølge, så oppstår det slike ”huller” med ledige trallenummer i arrayen ’status’.

**PS:** Dette med dyrepark på Gjøvik kan kanskje virke noe oppdiktet. Men, det er egentlig ikke så veldig komplisert: **Det er bare å sette opp et høyt og langt gjerde rundt hele byen !!!** Det eneste som da er urealistisk, er at det er nok med 30 traller !

**Lykke til !**

**frode@haug.com**

# Vedlegg: Halvferdig programkode (.tpl-fil)

```
// INCLUDE:
#include <iostream>           // cin, cout
#include <fstream>            // ifstream, ofstream
#include <cctype>              // toupper
#include <cstring>             // strcmp
using namespace std;

// CONST:
const int ANT_TRALLER = 30; // Antall traller som er til utleie.
const int NVNLEN = 40; // Max.lengde for en tekststreng.

// STRUCT:
struct tralle {              // Data om en tralle:
    char navn[NVNLEN];       // Dens navn.
    bool laant;               // Om utlånt (=true) eller ledig (=false).
};

// DEKLARASJON AV FUNKSJONER:
void skriv_meny();
char les_kommando();
int les_nr();
void utlaan() ;              // Oppgave 3a
void innlevering();          // Oppgave 3b
void les_fra_fil();           // Oppgave 3c
void skriv_til_fil();         // Oppgave 3d
void finn_person();           // Oppgave 3e
void utlaan2();               // Oppgave 3f

// GLOBALE VARIABLE:
tralle status[ANT_TRALLER+1]; // Array av tralle-struct'er
int ant_laant;                 // Antall traller som for øyeblikket er utlånt.

// MAIN:
int main() {
    char valg;
    les_fra_fil();             // Oppgave 3c
    skriv_meny();
    valg = les_kommando();

    while (valg != 'A') {
        switch(valg) {
            case 'U': utlaan(); break; // Oppgave 3a
            // case 'U': utlaan2(); break; // Oppgave 3f
            case 'I': innlevering(); break; // Oppgave 3b
            case 'F': finn_person(); break; // Oppgave 3e
            default : skriv_meny(); break;
        }
        valg = les_kommando();
    }
    return 0;
}
```

```

// DEFINISJON AV FUNKSJONER:
void skriv_meny() { // Skriver brukerens lovlige valg:
    cout << "\nFølgende kommandoer er lovlige:";
    cout << "\n\tU - Utlån av tralle";
    cout << "\n\tI - Innlevering av tralle";
    cout << "\n\tF - Finn ut om en gitt person låner";
    cout << "\n\tA - Avslutt";
}

char les_kommando() { // Leser, upcaser og returnerer ETT tegn:
    char ch;
    cout << "\nKommando: ";
    cin >> ch; // Leser.
    cin.ignore(); // Forkaster neste tegn (='\n').
    return toupper(ch); // Upcaser og returnerer.
}

int les_nr() { // Leser og returnerer et tall i arrayens indeks-intervall:
    int n;
    do {
        cout << "\n\tNummer (1-" << ANT_TRALLER << "): ";
        cin >> n; cin.ignore();
    } while (n < 1 || n > ANT_TRALLER);
    return n;
}

void utlaan() { // Låner ut (om mulig) en ledig tralle:
    // OPPGAVE 3a: Lag innmaten
}

void innlevering() { // Innleverer (om mulig) en tralle:
    // OPPGAVE 3b: Lag innmaten
}

void les_fra_fil() { // Leser inn dataene om tralleutlån fra fil og inn i datastrukturen:
    // OPPGAVE 3c: Lag innmaten
}

void skriv_til_fil() { // Skriver hele datastrukturen til fil:
    // OPPGAVE 3d: Lag innmaten
}

void finn_person() { // Finner ut om en gitt person har lånt noen tralle,
    // og skriver isåfall aktuelt trallennummer:
    // OPPGAVE 3e: Lag innmaten
}

void utlaan2() { // Annen versjon av oppgave 3a, som låner ut (om mulig)
    // FØRSTE ledige trallennummer:
    // OPPGAVE 3f: Lag innmaten
}

```