



Høgskolen i Gjøvik

E K S A M E N

FAGNAVN: Grunnleggende datakunnskap og programmering

FAGNUMMER: L 182 A

EKSAMENS DATO: 15. desember 1998

KLASSE: 98HINDA / 98HINDB / 98HINEA
98HDMUA / 98HDMUB / 98HINGA

TID: 09.00-13.00

FAGLÆRER: Frode Haug

ANTALL SIDER UTLEVERT: 8 (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 (15 %)

a) Internet:

- a1) Hva er et domenenavn ?
- a2) Hva er en nameserver ?
- a3) Forklar kort (max. 0.5 A4-ark) hva "Listserv" er.
- a4) Forklar kort (max. 0.5 A4-ark) hva "News/Usenet" er.

b) Datamaskinen (periferiutstyr / innmat):

- b1) Forklar kort (max. 0.5 A4-ark) om portene på en datamaskin.
- b2) Nevn seks inn-enheter og seks ut-enheter.
- b3) Hva er et "cluster" ?
- b4) Hva er de åtte hoveddelene på hovedkortet ?

c) Datasikkerhet:

- c1) Nevn fem områder for datakriminalitet.
- c2) Nevn fem eksempler på "fysiske sikringstiltak"
- c3) Hvilke seks hovedområder har vi for "håndtering av virus-problematikken" ?

d) Personvern / etikk:

- d1) Hvilke fem hovedoppgaver har Datatilsynet ?
- d2) Nevn fem nøkkelord for etiske verdier.

Oppgave 2 (25 %)

a) Hva blir utskriften fra følgende program:

```
#include <iostream>
using namespace std;

const int N = 5;

int main() {
    int i, j;
    for (i = 1; i < N; i++) {
        cout << char('F'+i) << ':';
        for (j = 0; j <= N; j++)
            cout << " " << i * j + 2;
        cout << '\n';
    }
    return 0;
}
```

b) Hva blir utskriften fra følgende program:

```
#include <iostream>
using namespace std;

const int STRLEN = 20;
enum tall { en, to, tre };

struct dyr {
    char navn[STRLEN];
    int antall;
};

dyr husdyr[3] = { {"Ku", 15}, {"Sau", 7}, {"Gris", 11} };

int main() {
    tall nr = to;
    int i;
    switch(nr) {
        case en: i = 0; break;
        case to: i = 1; break;
        case tre: i = 2; break;
        default: cout << "Alle husdyra er nå inne i fjøset!";
    }
    cout << "Av " << husdyr[i].navn
         << " har du " << husdyr[i].antall << " stk.\n";
    cout << "Du har";
    if (husdyr[0].antall < husdyr[2].antall)
        cout << " færre ";
    else cout << " likt eller flere ";
    cout << husdyr[0].navn << "er enn "
         << husdyr[2].navn << "er.\n";
    return 0;
}
```

c) Hva blir utskriften fra følgende program:

```
#include <iostream>
#include <cstring>
using namespace std;

char tekst[] = "ABCDEFGHijkl";

void skriv(char t[], int n, int m)
{ for (int i = n; i < m; i++) cout << t[i] << ' ';
  cout << '\n'; }

char hent(char t[], int nr)
{ return t[nr]; }

void rokk(char & a, char & b)
{ char c = a; a = b; b = c; }

int main() {
    skriv(tekst, 0, strlen(tekst));
    skriv(tekst, 2, 6);
    cout << hent(tekst, 4) << ' ' << hent(tekst, 8) << ' '
         << char(hent(tekst, 5)+2) << ' ' << int(hent(tekst, 7)) << '\n';
    rokk(tekst[4], tekst[6]);
    skriv(tekst, 2, 9);
    rokk(tekst[5], tekst[1]);
    skriv(tekst, 1, 6);
    return 0;
}
```

Oppgave 3 (60 %)

NB: Les *hele* teksten for denne oppgaven *nøy*e, før du begynner å besvare noe som helst. Studer spesielt vedlegg 1, slik at du får mer klarhet i hvordan det ferdige programmet skal fungere.

Gjør dine egne forutsetninger dersom du finner oppgaveteksten upresis eller ufullstendig. Gjør i så fall rede for disse forutsetningene i besvarelsen din.

Innledning

I denne oppgaven skal du lage et lite program som holder orden på en privatperson sine telefonnumre til ulike venner. Under kjøring av programmet skal dataene ligge i datamaskinens primærhukommelse. Det er maksimum plass til 200 ulike numre i dette kartoteket. For hver person lagres dens navn, telefonnummer, gateadresse og poststed (nummer og navn sammen). Som datastruktur *skal* du derfor bruke en array bestående av objekter (se vedlegg 1). Disse objektene er *ikke* sortert på noen verdi (f.eks. telefonnummer eller navn).

Legg merke til at vi bare lagrer *ett* nummer pr.person. Vi lagrer altså *ikke* både privatnummer, jobbnummer og/eller mobiltelefon. *Kun ett* av disse lagres.

Oppgaven

a) Deklarer/definer:

- **datamedlemmene i klassen** “Telefonnr”.
- **arrayen** du trenger for å representere datastrukturen din.
Husk at du skal ha plass til maksimum 200 telefonnummer.
- **hjelpevariabel** for å holde orden på hvor mange objekter i arrayen som hittil er brukt.

Bruk de ulike const'ene som er definert øverst i koden i vedlegg 1.

b) Lag resten av innmaten til funksjonen “void les(char txt[], char txt2[])”

Dette er en funksjon som skal skrive ut “txt” på skjermen og deretter et kolon. Så leser den brukerens tekstlige svar inn i “txt2”. Som du ser av vedlegg 1, så vil den deretter loope så lenge dette tekstsvaret kun er tomt (dvs. at brukeren kun taster ENTER/CR).

Utdypende:

Poenget med denne funksjonen er å tilkalle den hver gang man ønsker en tekst fra brukeren, f.eks. navn, gateadresse og poststed. Funksjonen kalles da med aktuell tekst som skal skrives på skjermen (“txt”), og den andre parameteren (“txt2”) som medsendes, vil ved retur fra funksjonen være fylt med brukerens svar. (Husk at “txt2” er en char-array som det ikke tas lokal kopi av inne i funksjonen, og at det dermed blir operert direkte på originalen.)

c) Lag funksjonene “void ny()” og “Telefonnr::les_data()” (Kommandoen ‘N’)

Dersom det ikke er plass til flere i kartoteket så gis en melding om dette. I motsatt fall så tas neste ledige objekt i arrayen i bruk, og dette blir bedt om å lese inn sine egne data. ”les_data()” skal bruke funksjonen laget i b) for å lese inn ikke-tomme verdier for navn, gateadresse og poststed.

NB: Du må gjerne bruke/tilkalle funksjonen fra b), selv om du ikke har kodet dens innmat, bare du skjønner hvordan den skal brukes.

d) Lag funksjonene “void oversikt()” og “Telefonnr::skriv_data(int n)” (Kommandoen ‘O’)

Dersom kartoteket er tomt, så gis en melding om dette. I motsatt fall gås det gjennom alle de brukte objektene i arrayen. For hvert av dem, skrives det ut indeksnummer, telefonnummer og navn på en linje, mens gateadresse og poststed skrives samlet på en annen (med komma imellom).

Det er funksjonen ”skriv_data” inni hvert objekt som utfører dette siste.

e) Lag funksjonen “bool slett(int n)” (Kommandoen ‘S’)

Som inn-parameter så mottar denne funksjonen indeksen (“n”) til objektet i arrayen som ønskes fjernet/slettet. Dersom denne verdien er utenfor intervallet av indekser som er i bruk, så returnerer funksjonen ’false’. I motsatt fall så flyttes det siste objektet inn på den plassen der det som skulle fjernes lå, antall brukte objekter i arrayen telles ned med 1, og funksjonen returnerer ’true’.

NB: Legg merke til hvordan denne funksjonen “slett(...)” brukes i koden i vedlegget.

f) Lag funksjonene “int finn(char nv[])” og “Telefonnr::lik_navn(char nv[])” og “Telefonnr::hent_telefonnummer()” (Kommandoen ‘F’)

Funksjonen leter seg fram (bl.a. vha. ”lik_navn”) i arrayen etter et objekt som har navnet “nv”. Dersom den lykkes i å finne dette, så får den tak i vedkommendes telefonnummer (vha. ”hent_telefonnummer”) og returnerer dette. I motsatt fall så returneres bare “0”.

NB: Legg merke til hvordan denne funksjonen “finn(...)” og dens retur-verdi brukes i koden i vedlegget.

g) Lag funksjonene “void les_fra_fil()” og “Telefonnr::les_fra_fil(istream * inn, int nr)”

I vedlegg 2 ser du helt konkret hvordan filen “KARTOTEK.DTA” eksempelvis kan se ut. Hver post består av fire felter (telefonnummer, navn, gateadresse og poststed), og disse er lagret på hver sin linje. Les alt på denne filen slik at den (i starten av programmet) tomme data-strukturen fylles med verdiene fra filen. Husk å telle opp antall objekter som er i bruk av arrayen!

Lykke til, og ring riktig mye til venner i julen !
FrodeH

Vedlegg 1: Halvferdig programkode (Fil: C++ \ EXTRAMEN \ EX_H98_3.TPL)

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

// CONST:
const int STRLEN = 60;     // Max.lengde for en tekststreng.
const int MAXTLF = 200;    // Max.antall i kartoteket.

class Telefonnr {         // KLASSE:
private:
    // Oppgave 3a: Deklarer data-innmaten
public:
    // Lag innmaten til disse ifm:
    void les_data();        // Oppgave 3c
    void skriv_data(int n); // Oppgave 3d
    bool lik_navn(char nv[]); // Oppgave 3f
    int hent_telefonnummer(); // Oppgave 3f
    void les_fra_fil(istream * inn, int nr); // Oppgave 3g
};

void skriv_meny(); // DEKLARASJON AV FUNKSJONER:
void les(char & ch);
void les(char txt[], char txt2[]);
void ny();
void oversikt();
bool slett(int n);
int finn(char nv[]);
void les_fra_fil();

// Oppgave 3a: Definer datastrukturen

int main() { // HOVEDPROGRAMMET:
    char kommando; // Brukerens valg/ønske/kommando.
    int nr;         // Indeksen som skal slettes (se case 'S').
    int nummer;     // Indeksen til funnet telefonnummer (se case 'F').
    char navnet[STRLEN]; // Navnet tilhørende ønsket tlf.nr. (se case 'F').

    les_fra_fil(); // Oppgave 3g
    skriv_meny();
    les(kommando); // Leser brukerens valg/ønske/kommando.
    while (kommando != 'Q') { // Så lenge ikke avslutte (quit):
        switch(kommando) {
            case 'N': ny(); break; // Oppgave 3c
            case 'O': oversikt(); break; // Oppgave 3d
            case 'S': cout << "\tHvilken indeks vil du slette: "; cin >> nr;
                if (slett(nr)) // Oppgave 3e
                    cout << "\n\tDenne indeksen er nå slettet.";
                else cout << "\n\tIndeksen utenfor relevant område!"; break;
        }
    }
}
```

```

    case 'F':    les("Navn", navnet);        // Oppgave 3b
                nummer = finn(navnet);      // Oppgave 3f
                if (nummer != 0)
                    cout << "\n\tTelefonnummeret er: " << nummer;
                else cout << "\n\tIngen i kartoteket har dette navnet!"; break;
    default:    skriv_meny();
}
les(kommando);
} // Burde ha skrevet datastrukturen tilbake til fil.
// skriv_til_fil(); // Men dette er ikke en del av denne eksamensoppgaven.
return 0;
}

// DEFINISJON AV FUNKSJONER:
void skriv_meny() {
    cout << "\n\nFølgende kommandoer er lovlige:"
        << "\n\tN - Nytt telefonnummer inn i kartoteket."
        << "\n\tO - Oversikt over hele kartoteket skrives til skjermen."
        << "\n\tS - Slette alt på en gitt indeks i kartoteket."
        << "\n\tF - Finn telefonnummeret for en gitt person."
        << "\n\tQ - Quit/avslutt.";
}

// Oppdaterer direkte den 'ch' er et alias for
void les(char & ch) { // (da denne er referanseoverført).
    cout << "\n\nKommando: "; cin >> ch; // Leser brukerens valg/ønske/kommando.
    ch = toupper(ch); cin.ignore(); // Gjør om til STOR bokstav og forkaster neste tegn ('\n').
}

void les(char txt[], char txt2[]) {
    do {
        // Oppgave 3b: Lag resten av innmaten
    } while (strlen(txt2) == 0); // Looper så lenge kun ENTER/CR.
}

void ny() {
    // Oppgave 3c: Lag resten av innmaten
}

void oversikt() {
    // Oppgave 3d: Lag resten av innmaten
}

bool slett(int n) {
    // Oppgave 3e: Lag resten av innmaten
}

int finn(char nv[]) {
    // Oppgave 3f: Lag resten av innmaten
}

void les_fra_fil() {
    // Oppgave 3g: Lag resten av innmaten
}

```

Vedlegg 2: Eksempel på innhold på filen “KARTOTEK.DTA”

11111111
Arne Arvik
Astrupgate 1
1111 Arendal
22222222
Bengt Bruvik
Balkegate 2
2222 Bergen
33333333
Cecilie Carlsen
Camilla Colletsgate 3
3333 Copenhagen
44444444
Doris Day
Dalbakkveien 4
4444 Drammen
55555555
Eirik Eklund
Einar Skjæråsgate 5
5555 Elverum
66666666
Frank Fredriks
Fiolveien 6
6666 Flekkefjord
77777777
Gurine Gråthen
Gamlegutua 7
7777 Grimstad
88888888
Harald Hårfagre
Hafsrjordgate 8
8888 Holmestrand
99999999
Iselin Ingvaldsen
Ilterveien 9
9999 Islandia

Formatet er altså:

<Telefonnummer>
<Navn>
<Gateadresse>
<Poststed>