

Datorövning 1

- A) Kör igång Microsoft Visual C++-miljön. Välj File | New | Project från huvudmenyn och skapa ett nytt Win32 Console Project med namnet Mittnamn. Skapa en ny katalog, Oop, genom att vid Location bläddra till katalogen M:\PCSA\ och där skapa en ny katalog genom att använda musens högerknapp och i menyn välja, Nytt. Ditt projekt hamnar nu i katalogen M:\PCSA\Oop\Mittnamn\.. Glöm inte att välja ett tomt projekt med empty i Applicationfönstret som dyker upp.

Välj från huvudmenyn View | Solution Explorer och klicka med högerknappen på Source Files, välj Add och sedan Add New Item. I fönstret som dyker upp väljer du C++ File och ger filen namnet Mittnamn.cpp. I denna fil ska du skriva ett program som läser in ditt namn (både för- och efternamn) och skriver ut namnet ungefär mitt på skärmen. Kompilera genom att med högerknappen klicka på Mittnamn.cpp i Solution Explorer och välja Compile. Länka genom att från huvudmenyn välja Project Build och kör programmet med Start från Debug-menyn. Använd funktioner från iostream och conio för in- och utmatning och från MSViscon (kan laddas ner från kurssidån) för skärmhantering.

MSViscon.cpp och MSViscon.h tar du med i projektet genom att med högerknappen på Source Files välja Add och sedan Add Existing Item och på samma sätt med högerknappen på headerfiles för MSViscon.h. Kör programmet genom att från huvudmenyn välja Debug Start. Testa både `cin >>` och `cin.getline` vid inläsningen. Använd hjälpfunktionen (F1) för att hitta lämpliga funktioner samt se hur de anropas. När du har testat programmet färdigt stänger du det med File | Close Solution.

-
- B) Skapa ett projekt Forkorta, som läser in täljare och nämnare till ett bråk och skriver ut detta förkortat. För att förkorta använder du funktionen `sgd` nedan som returnerar största gemensamma delaren till två tal. Skriv in funktionen före main i samma fil.

```
int sgd(int m, int n)
{
    int rest;

    while ( n != 0 )
    {
        rest = m % n;
        m = n;
        n = rest;
    }
    return m;
}
```

Kompilera länka och kör programmet. Kör också programmet radvis i debuggern genom att använda F10 eller F11. Ställ muspekaren på någon variabel och kolla in värdet på variabeln.

- C) Skapa ett nytt tomt projekt Carmain och ta fram Solution Explorer där du med högerknappen på Source Files adderar en ny fil Car.cpp till projektet och på samma sätt på Header Files adderar en ny fil Car.h enligt nedan. Implementera funktionerna i filen Car.cpp och kompilera separat. Skriv sedan ett huvudprogram i filen Carmain.cpp som läser in två bilar och skriver ut dem i ordning med den lägsta effekten först.

```
// Specifikation av klassen CAR - Car.h

class CAR
{
    private :
        char regnr[7];    // Registreringsnr
        float power;     // Effekt
        int nrcyl;       // Antal cylindrar

    public :
        void las();
        void skriv();
        bool is_less_power(CAR c);    // Sant om objektet har lägre
                                      // effekt än c
};
```

- *D) Klassen RTAL nedan ska avbilda bråk. Implementera funktionerna i filen Rtal.cpp. Se till att varje räkne-operation returnerar ett förkortat bråk. Skriv sedan ett huvudprogram Rtalmain.cpp, som inkluderar Rtal.h och läser in tre bråk a, b och c och skriver ut värdet av uttrycket $a + b*c$. Kompilera, länka och kör.

```
// Specifikation av klassen RTAL -- Rtal.h

class RTAL
{
    private :
        int t;
        int n;

    public :
        void las();
        void skriv();
        void forkorta();
        RTAL add(RTAL rt);
        RTAL sub(RTAL rt);
        RTAL mul(RTAL rt);
        RTAL div(RTAL rt);
};
```

Datorövning 2

- A) Skriv en funktion `swap`, som tar två flyttal som parametrar och byter plats på dessa. Skriv först funktionen med pekare som parametrar och sedan med referensparametrar. Skriv ett enkelt huvudprogram som läser in två tal, kör `swap` och skriver ut talen.
-

- B) Skriv en liknande överlagrad funktion `swap` (samma namn) som istället tar två strängar som parametrar och byter plats mellan dessa. Använd samma huvudprogram som ovan där du efter talen fortsätter att läsa in två strängar (max 80 tecken), kör `swap` och skriver ut strängarna. Kontrollera med debuggern vilken funktion som körs.
-

- C) Skriv en funktion som tar en vektor av tärningskast och antalet kast som parametrar och returnerar medelvärdet av kasten. Låt funktionen ha defaultvärdet 1000 på antalet kast.

Skriv ett huvudprogram som frågar efter hur många kast som ska göras och som dynamiskt skapar en vektor som exakt rymmer kasten. Anropa sedan funktionen som beräknar medelvärdet och skriv ut resultatet

- D) Gör om huvudprogrammet i uppgift C ovan så att antalet kast kan skickas som parameter till `main`. Anger man bara en parameter så ska 1000 kast göras. Kör först programmet i Visual studio miljön och sedan i Console miljön. Observera att alla parametrar till huvudprogrammet är strängar och måste göras om av ditt program till heltal med exempelvis funktionen `atoi`.
-

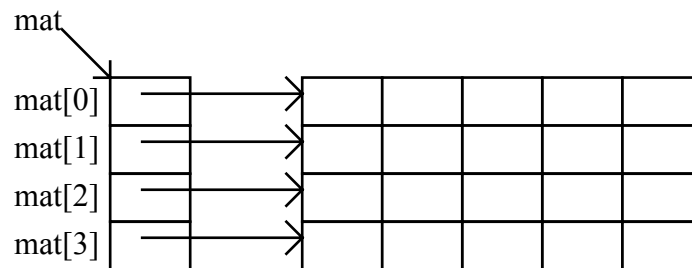
- *E) Implementera medlemsfunktionerna till klassen `VEKTOR` med specifikationen :

```
// Specifikation av klassen VEKTOR -- Vektor.h
class VEKTOR
{
    private :
        float *vp;
        int storlek;

    public :
        void skapa(int s);           // skapa dynamiskt ett VEKTOR-
                                    // objekt med storlek s
        void tabort();              // avallokera ett VEKTOR-objekt
        void las();                 // läs in elementen
        void skriv();               // skriv ut elementen
};
```

samt skriv ett huvudprogram som skapar en vektor med tre element och skriver ut den på skärmen innan den tas bort.

- **F) En matris kan ses som en vektor av vektorer eller pekare som exempelvis matrisen med 4 rader och 5 kolumner nedan :



```
// Specifikation av klassen Matris -- Matris.h
```

```
class MATRIS
{
    private:
        float **mat;
        int rad, kol;

    public:
        void skapa(int r, int k);
        void tabort();
        void las();
        void skriv();
};
```

I funktionen skapa allokerar man först r st pekare till flyttal och sedan kör man en loop och allokerar k st flyttal för varje rad. I funktionen tabort avallokeras först varje rad av flyttal och därefter tas pekarna bort.

Skriv ett huvudprogram som skapar en matris med 2 rader och 3 kolumner och skriver ut den på skärmen innan den avallokeras.

Datorövning 3

- A) Skriv ett Windows-program som visar texten Hej mitt i ett fönster om du trycker på knappen Show/Hide och som gömmer texten då man trycker på samma knapp en gång till. Välj File | New | Project, Windows Forms Application och sätt namnet Datorovn3A på projektet. Här får du en hel del av ditt program färdigt. Du får ett huvudformulär (Huvudfönster) Form1 där du kan placera olika komponenter i Design-fönstret Form1.h(design). Huvuddprogrammet, Form1.cpp, är färdigt och den kod som du ska skriva skriver du i filen Form1.h.

Markera formuläret Form1 i Form1.h(design) fönstret och sätt i Property-fönstret, som du får fram genom att klicka med högerknappen i Formuläret Form1, property (egenskapen) Text till "En hälsning från", följt av ditt namn. Välj View | Toolbox och klicka på komponenten Label (rubrik) i Window Forms och dra ut och placera en Label på formuläret Form1 i design-fönstret. Sätt property Namn till HejLabel och sätt property Text till Hej med största möjliga font som du väljer genom att klicka på de tre punkterna hos property Font. Välj sedan komponenten Button från Toolbox menyn och placera den under HejLabel. Ge den namnet ShowHideButton och texten Show/Hide. Markera knappen Show/Hide och klicka på events och dubbelklicka på den vita ytan vid händelsen click. Nu har du fått en händelsefunktion i filen Form1.h där du kan skriva vad som ska hända när du klickar på knappen Show/Hide. Skriv in koden

```
if (HejLabel->Visible)
    HejLabel->Hide();
else
    HejLabel->Show();
```

Provkör och testa. Du kanske vill att Hej inte ska synas förrän du har tryckt på knappen. Det kan du fixa genom att antingen ändra property Visible eller också skriva kod i konstruktorn i Form1.h.

- B) Skriv ett program som istället för att visa Hej i fönstret visar en bitmapbild. Istället för en Label ska du använda en PictureBox. Denna komponent plockar du ner från Toolboxen och ger den namnet FotoPictureBox. För att ladda ner bilden klickar du på Picture i property. Klicka på de tre punkterna och leta med browsern fram den bild som du vill visa. För att visa bilden byter du ut koden som visade Hej till

```
if (FotoPictureBox->Visible)
    FotoPictureBox->Hide();
else
    FotoPictureBox->Show();
```

- C) Skriv ett program som visar en TextBox-komponent innehållande ditt namn istället för en Label som i uppgift A. Namnet i textboxen ska dyka upp då man klickar på Show/Hide och försvinna då man klickar en gång till. Använd knappens Click-händelse där du anropar MyTextBox enligt:

```
if (MyTextBox->get_Text()->Length == 0)
    MyTextBox->set_Text("Kalle Anka");
else
    MyTextBox->set_Text("");
```

- *D) Skriv ett program som har en knapp Show som när man klickar på den visar en MessageBox med en hälsningstext. För att visa en MessageBox skriver du

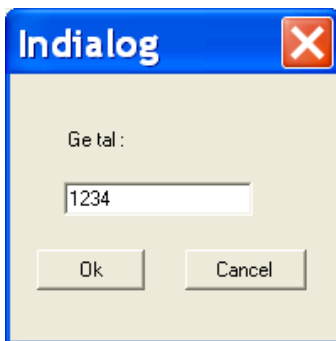
```
MessageBox::Show("Hej", "Hälsning från Kalle Anka");
```

Eftersom Show är en klassmetod , alltså en metod som är gemensam för alla objekt i klassen, anropas den med klassnamnet. Använd OK-knappen på MessageBoxen för att den ska försvinna.

- **E) Skriv ett program som via en meny kan läsa in ett heltal, skriva ut talet och avsluta programmet. Grafiskt användargränssnitt (GUI) enligt:



Läs ska ge en egentillverkad inmatningsbox, som du adderar till projektet med Add New Item och väljer Item till WindowsForm, enligt:



Lägg sedan till knappar (Ok och Cancel), textbox och Label (Ge tal) och använd ShowDialog på Indialogfönstret så att du kommer åt det inmatade värdet. Skriv ska ge en MessageBox enligt:

