Filer i C++

- Egna filobjekt (jämför cin och cout)
  skapa filpekare
  öppna fil och ange ändamål
  (text / binär - läsning / skrivning)
  Läsa och skriva textfiler
  Läsa och skriva binärfiler

- Genomgång av övningstenta

Filer i C++

Deklareras i **fstream.h**

```
ifstream     // för läsning från filer
ofstream     // för skrivning till filer
fstream      // för skrivning och läsning
```

Skapa filpekare

```
ifstream fin;
ofstream fout;
fstream fio;
```

Öppna filer i C++

Öppna filer med **open()**

```
// Deklaration av filpekare
ifstream fin;
ofstream fout;

// Öppna textfil för läsning
fin.open( ”SLASK.TXT” );
// Öppna binärfil för skrivning
fout.open( ”SLASK.DAT”, ios::binary );
// eller med en fstream
fstream fil;
fil.open( ”SLASK.DAT”, ios::binary | ios::out);
```

Filens användning

Användning bestäms vid öppningen

```
ios::in          // för läsning
ios::out         // för skrivning
ios::app         // lägg till i slutet
ios::binary      // binärfil
```

Lägena kan kombineras med | (bitvis OR)

```
ios::in | ios::out // läsning och skrivning
ios::in | ios::bin // binärfil för läsning
```

Deklarera och öppna filer

Man kan öppna filen vid deklarationen

```
// Öppna en textfil för läsning
ifstream intext( ”minfil.txt” );
// Öppna en binärfil för skrivning
ofstream utdata( ”calc.dat”, ios::binary );
// Öppna binärfil för läsning och skrivning
fstream databas( ”dbase.dat”,
                 ios::binary
                 | ios::out
                 | ios::in);
```

Direktåtkomst

Man kan flytta sig till godtycklig plats i filen

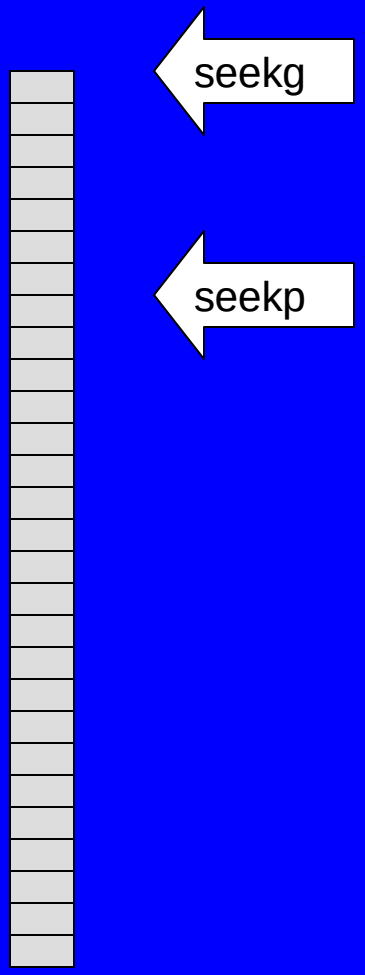
Med hjälp av två interna pekare i filen

`seekp()`	för skrivning
`seekg()`	för läsning

Normalt pekar de på första byten
    då filen öppnas

Stega framåt eller bakåt:

```
seekp( steg, startpunkt );
```



Stega framåt eller bakåt:

Steget - antal bytes anges som en **`long int`**

Startpunkten är början, slutet eller nuvarande position

```
seekp( steg, startpunkt );
```

Startpunkter anges med

```
ios::beg     // från början
ios::end     // från slutet
ios::cur     // från nuvarande position (current)
```

```
fin.seekp( 0L, ios::beg ); // Gå till början
fin.seekp( 20, ios::cur ); // 20 bytes framåt
fin.seekp( -1, ios::end ); // 1 före sista
fin.seekp( -sizeof( Komplex ),ios::cur);
```

Testa filen

Lämpligt att testa att det fungerar

```
if(! infil) {
   cout<<”Kan inte öppna TESTFIL.DAT”;
   exit(1);
}
```

Test på infil returnerar NULL (’\0’) om det inte gick att öppna eller skapa filen.

Skriva och läsa textfiler

Med utmatnings och inmatningsooperatorerna

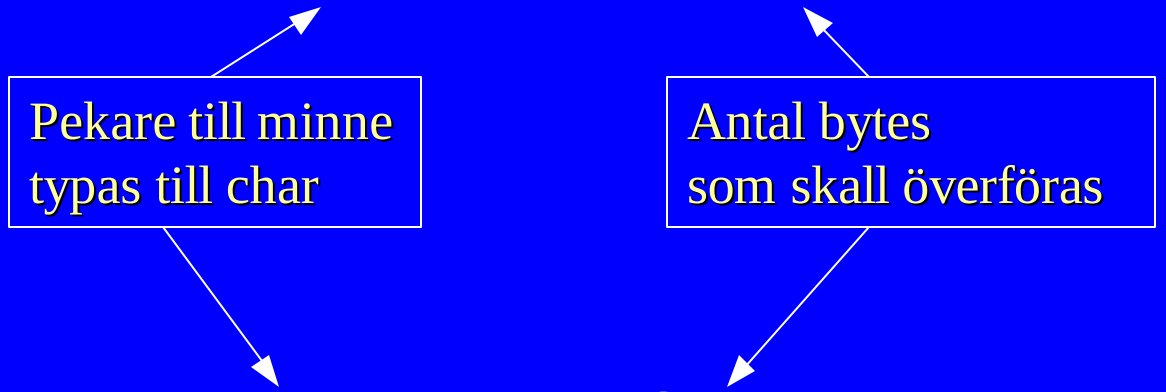
```
ostream utfil(”data.txt”);
float a = 123.8, b = 2E-3;
utfil << a << b;
utfil.close();

istream infil(”data.txt”);
char s[20];
infil >> s;
infil.close();
```

Skriva och läsa binärfiler

Poster (struct) sparas lämpligen på binärfiler

Skriv och läs med **`write()`** och **`read()`**

```
ofstream  utfil("TESTFIL.DAT", ios::binary);
PostTyp post, *p = post;
dbase.write((char*) p, sizeof( PostTyp ));
```

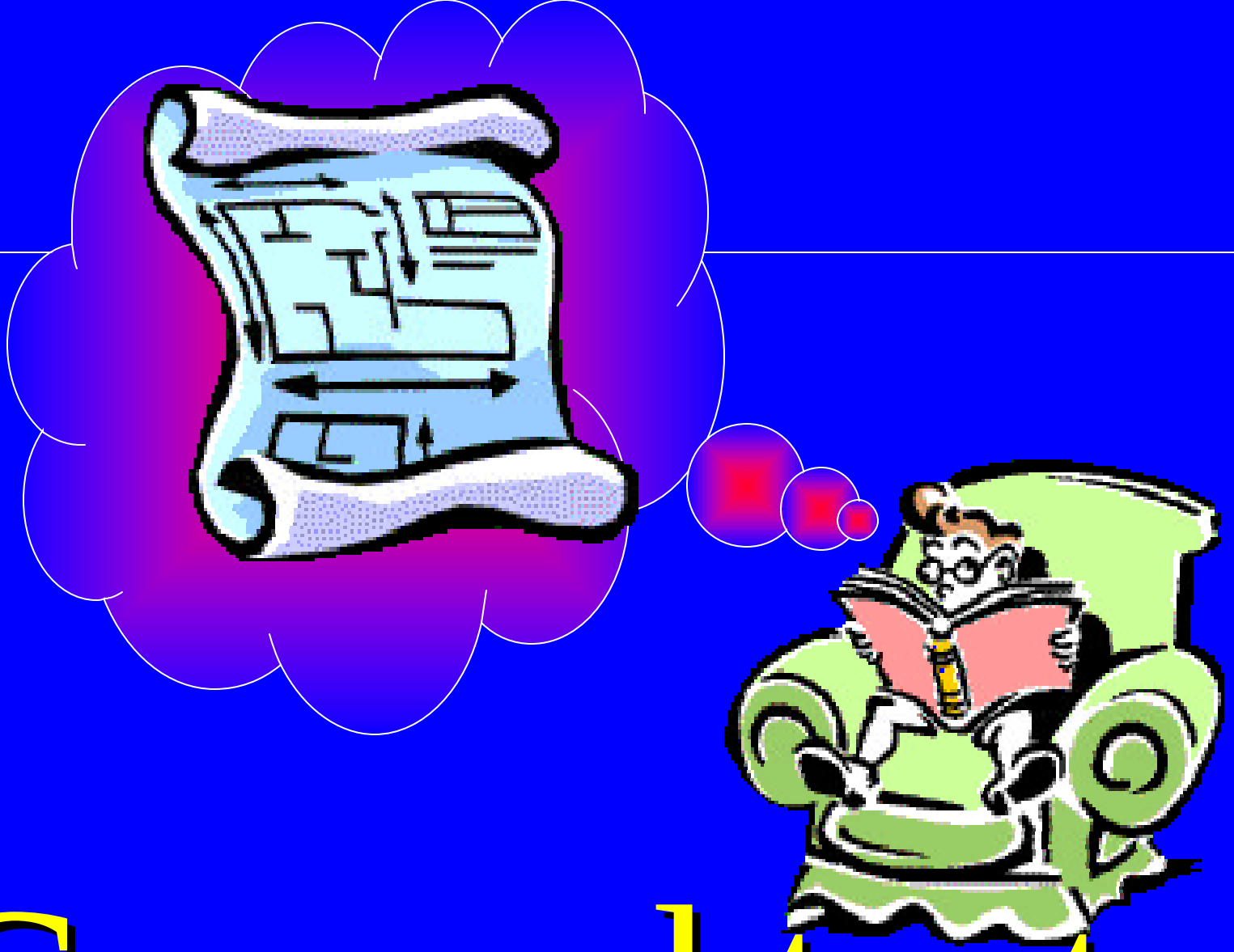



```
dbase.read((char*) p , sizeof( PostTyp ));
```

Skriv en post på fil

```
struct PostTyp {    // Typdefinition
       char enamn[20];
       char fnamn[20];
       char tel[15];
     };
void savepost( PostTyp *post ){
   fstream dbase(filnamn, ios::binary |ios::in
                          |ios::out);
   PostTyp *p = post;
   dbase.seekp(0L,ios::end);
   // Skriv ut post
   dbase.write((char*)p,sizeof( PostTyp ));
}
```

dbase.cpp



Gammal tenta