

Programmering i C++
EDA623
7.5 hp

Tecken och texter

char

```
char ch = 'A';
```

teckenfält = strängar i C

```
char str[] = "123";
```

standardklass i C++

```
string s = "C++";
```

8-bitars tecken (ej 16-bitars som i Java)

```
char c1, c2;  
c1 = 'A';  
c2 = 106; // Samma som 'j'  
cout << c1 << c2 << '!' << endl;
```

Två olika 8-bitars-tabeller

- ISO-8859-1 (Latin-1) (Se Appendix B)
 - Windows code page 28591 (ISO-8859-1) (Latin-1)
 - Windows code page 1252 (Windows Latin-1)
- 850 (IBM-ASCII): Används bara i DOS

Ändra kodning i konsolfönstret

```
#include <iostream>
#include <cstdlib>

using namespace std;

int main()
{
    system("chcp 1252 >nul 2>nul");

    char ch = '7';
    int code, val;
    code = ch;
    val = ch - '0';
    cout << "Siffran " << ch << " har koden " << code
         << " och värdet " << val << endl;
}
```

Översätt mellan program och konsoll

```
#include <iostream>
#include "iodos.h"

using namespace std;

int main()
{
    dos_console();

    char ch = '7';
    int code, val;
    code = ch;
    val = ch - '0';
    cout << "Siffran " << ch << " har koden " << code
         << " och värdet " << val << endl;
}
```

Escape-sekvenser (s. 71 i boken)

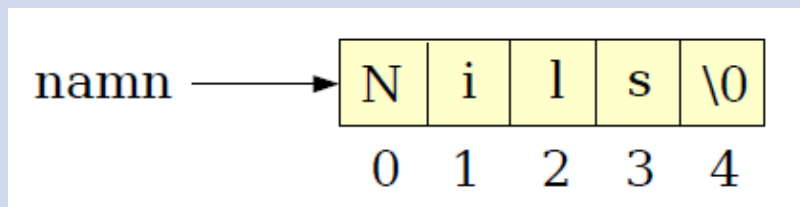
```
\n    (asciikod 10 = new line, line feed)
\a    (asciikod  7 = bell)
\t    (asciikod  9 = tab)
\r    (asciikod 13 = carriage return)
\b    (asciikod  8 = backspace)
\nnn  (tecken med asciikod nnn (oktalt))
\xnn  (tecken med asciikod nn (hexadecimalt))
\\    (backslash)
\"    (citationstecken)
\'    (apostrof)
```

Teckenfält ("C-strängar")

Lagring av teckenfält

- Fält bestående av char
- Avslutas alltid med tecknet \0

```
char namn[] = "Nils";
```



```
cout << namn[1] << namn[3]; //is
```

In- och utmatning av tecken

```
cout << c           // Skriver ut tecknet c
cout.put(c)        // Skriver ut tecknet c

cin >> c           // Läser in nästa icke-vita tecken till c
cin.get(c)         // Läser in nästa tecken till c

cin.peek()         // Returnerar nästa tecken som förblir oläst
cin.ignore()       // Läser nästa tecken men slänger det
```


Exempel 1

```
#include <iostream>
using namespace std;
int main(){
    char namn[30];
    cout << "Vad heter du? ";
    cin >> namn;
    cout << "Hej, " << namn << "!" << endl;
}
```

Vid körning:

```
Vad Heter du? Lars Olof Persson
Hej, Lars!
```

Byt ut `cin >> namn` mot `cin.getline(namn,30)`.

Exempel 2

```
#include <iostream>
using namespace std;
// Kryptiskt program
char kod[33]="qwertyuiopasdfghjklzxcvbnm012345";
int main()
{
    char c;
    cout << "Avslutning sker med Ctrl-z" << endl;
    while (cin.get(c)) {
        cout << kod[c%32];
    }
    cout << endl;
}
```

Exempel 3

```
#include <iostream>

using namespace std;

int main()
{
    string s;
    int tal = 0;
    cout << "Ange ett tal: ";
    cin >> s;
    for (char ch : s) { // C++11
        tal = 10 * tal + ch - '0';
    }
    cout << "Talets värde: " << tal << endl;
}
```

In- och utmatning av tecken

```
#include <cstring> // Bibliotek att inkludera

strcpy(s,t) // Kopierar t till s
strncpy(s,t,n) // Variant med max n tecken

strcat(s,t) // Lägger till t till slutet av s
strncat(s,t,n) // Variant med max n tecken

strlen(s) // Returnerar längden av s

strcmp(s,t) // Jämför s och t
strncmp(s,t,n) // Variant med max n tecken
// s<t, s==t, s>t ger resp. <0, =0, >0
```

Initiering och tilldelning

Initiering

```
char s[4]="abc"; // ok
char s[]="abc"; // ok
char s[3]="abc"; // Inte ok (pga \0)
```

Tilldelning

```
s = "def"; //Felaktigt!
s[0]='d'; s[1]='e'; s[2]='f'; // ok

// Bättre variant:
strcpy(s, "def"); // kräver #include <cstring>
```

Enklare hantering än med teckenfält

```
#include <string>

string s1("abc"), s2="def", s3;

s3 = "ghij";
s3 = s1 + s2 + s3 + "klmnop";
s3 += "qrstuvwxyz";
```

Operatörer på string

```
#include <string>

s = t;           // Tilldelning
s.assign(t);    // Tilldelning
s = s + " " + t; // Konkatenering
cout << s << t; // Utskrift
cin >> s;       // Inmatning av ord
getline(cin, s); // Inmatning av hel rad
s < t, s == t    // Jämförelse
s[k]            // Indexering, hämta eller sätta värdet
s.at(k)        // Indexering med indexkontroll
```

Funktioner (metoder) i string

`s.substr(k,n)`

`s.erase(k,n)`

`s.clear()`

`s.size()`

`s.append(t)`

`s.insert(k,t)`

`s.replace(k,m,t)`

`s.find(t)`

std::vector

```
int fib[] {1,1,2,3,5,8,13,21};  
vector<int> vfib(begin(fib), end(fib));  
for (int i : vfib) {cout << i << " ";}  
cout << endl;  
  
vector<int> v2 {1,2,3};  
  
char str[] = "Hello World!"; // kräver =  
vector<char> v3(begin(str), end(str));
```

std::array

```
array<char,13> a1 {"Hello Again!"}; // extra {}  
for (char ch : a1) {cout << ch << " ";}  
cout << endl;
```