

SNABBGUIDE MED KOMMANDON OCH STRUKTURER I C++

```
 dessa paket måste alltid finnas (om utmatning och inmatning ska fungera)
#include <iostream>
using namespace std;
```

```
 huvudfunktion måste alltid finnas
int main()
{
    system("PAUSE");
    return 0 ;
}
```

```
 variabler som används måste deklarerars t.ex.
int  heltal, antal, nr ;
double  decimaltal, tal1, tal2, tal3 ;
```

```
 ibland måste vi initiera variablerna för att det inte ska bli fel t.ex.
int  summa = 0, produkten = 1 ;
double  domarsumma = 0.00 ;
```

```
 ska vi använda en variabel i programmet som får ett värde vid deklarationen och sen får inte värdet ändras under
programmets gång deklarerar vi den som en konstant t.ex.
const double pi=3.14 ;
```

```
 utmatning på skärm med avslutande radbrytning skrivs
cout<<"Hej svejs!"<<endl;
```

```
 inmatning från tangentbord till program skrivs
cin >> tal;
```

```
 behöver du matematiska funktioner måste du inkludera mattembiblioteket du
har då tillgång till t.ex.      pow(x,y), sqrt(x), sin(x), cos(x), tan(x), exp(x)
#include <math.h> //alt. <math>
```

```
 heltaldivision kapar decimalerna t.ex 7/3, ger 2 till svar
%-tecknet(modulo) används för att ge resten 7%3,ger 1 till svar
Vill du erhålla decimaltal vid division, kontrollera täljare och nämnare.
```

```
 Ibland måste vi kunna välja olika vägar beroende på om vissa villkor
stämmer eller inte
if (uttryck)
{
    en eller flera satser
}
else if (annat uttryck)
{
    en eller flera satser
}
else
    vad ska göras om både if och else if är fel
```

Operator	Betyder	Exempel
<	Mindre än	if (a<20)
>	Större än	if (b>100)
==	Lika med	if (produkt==500)
!=	Skiljt från (inte lika med)	if (x!=28)
<=	Mindre än eller lika med	if (a<=2000)
>=	Större än eller lika med	if (b>=12)

Operator	Betyder	Exempel
&&	och	if (a>20) && (a<30)
	eller	if (a==0) (b<0)
!	inte	if (!(a>0))

Vill vi upprepa några satser ett känt antal gånger använder vi oss av for loopen.

```
for (initiering av startvärde; villkor; upp/ner-räkning )
{
    en eller flera satser;
}
```

t.ex.

```
for (int a=0;a<20;a++)
{
    cout<<a<<endl;
}
```

vet vi inte när vi börjar loopen hur många varv som ska upprepas använder vi hellre while-loopen där vi kan ställa fler villkor

```
while ( uttryck som ska vara sant om det innanför klamrarna ska göras)
{
    en eller flera satser;
}
```

t.ex.

```
while (totalt_belopp < onskat)
{
    ant_dagar++;
    dagslon = dagslon*2;
    totalt_belopp = totalt_belopp + dagslon ;
    cout << " Dag " << ant_dagar ;
    cout << "          Dagslön = " << dagslon ;
}
```

behöver vi variabler som är indexerade använder vi oss av arrayer, deklarerar t.ex.

```
int talserie[25] , double decimalserie[50]
```

vill vi att alla tal ska vara 0 vid start

```
for (int a=0;a<25a++)
    talserie[a] = 0;
```

vill vi mata in domarsiffror

```
int domarVarden[10];
for ( int nr=1 ; nr<=antalDomare ; nr++ )
{
    cout << "Mata in ditt värde" << endl;
    cin >> domarVarden[nr];
}
```

Vill vi formatera utmatningen på skärmen lägger vi till

```
#include <iomanip.h> //alt <iomanip>
ock kan då använda
cout << "Dag " << setw(6) << antDagar ; // sätter bredd på utmatningsfältet
cout << setprecision(2)<<setiosflags(ios::fixed)<< decimalTal <<endl;
// sätter antalet decimaler
```

```
srand (time(0)); // startar slumpvalsfrö
rand(); // kastas ger felaktigt värde
tall=(int)(rand()/32767.01*100); // ger heltal 0-100, 100?
```