C++ for begyndere

Jeg har lavet denne lille artikelserie som giver en god start til at lære c++. Artiklen er stor da det er 2 artikler der er slået sammen.

Skrevet den **03. Feb 2009** af **visualdeveloper** I kategorien **Programmering / C/C++** | ★★★☆☆

Der er to af disse artikler, fordi jeg har slået nr 1 og nr 2 sammen i en artikel. Jeg vil ikke slette den ene af hensyn til dem der har betalt for den !

Lad os starte med c++.

Når vi skal skrive et program i c++ skal vi have to stykker software:

1. en editor
2. en compiler

Editoren & Compileren
Til at skrive al teksten/kildekoden og gemme det som en .cpp fil skal vi bruge en editor. En simpel editor kunne fx være notesblok. Der skriver man bare programmet og så gemmer man det som en .cpp fil. En anden god editor som jeg godt kan anbefale er TextPad som kan downloades her: http://www.textpad.com/index.html.
Men....
Den aller simpleste vej er et program hvor man både kan skrive programmet og compile det. Sådan et program kaldes IDE eller Integrated Development Environment. Et IDE program jeg kan anbefale er Dev-C++ som kan downloades her: http://www.bloodshed.net/devcpp.html.
Det er også Dev-C++ jeg vil tage udgangspunkt fra i denne artikel.

Dit første program
I om ikke alle så i hvert fald 99.9 % af artikler/bøger om c++ starter man med eksemplet "Hello World", så hvorfor ikke også starte med det her.

```
// Mit første C++ program

#include <iostream>

using namespace std;

int main ()
{
  cout << "Hello World!";
  return 0;
}
```

Sæt dette ind i editoren og gå ind i menuen Kompiler --> Kompiler og kør.

Du kan måske lige nå at ane en sort boks komme op og lukke igen.
Det var dit program.

Hvis du gerne vil nå at se dit program skal vi lige indsætte en "pause"

```
// Mit første C++ program med pause

#include <iostream>

using namespace std;

int main ()
{
  cout << "Hello World!";
  std::cin.get();
  return 0;
}
```

Nu skulle du gerne få et console vindue (DOS-vindue) hvor der står
Hello World
Press any key to continue...

Lad os kigge nærmere på programmet

// Mit første C++ program - dette er bare en kommentar som du til hver en tid kan indsætte i dit program. Bare lav to skråstreger (//) og skriv så din kommentar bagefter.

#include <iostream> - alle sætninger der begynder med #include fortæller at programmet skal inkludere iostream som indeholder dekarationer på c++'es standart output bibliotek, som vi bruger i vores program (cout).

using namespace std; - dette er et namespace, og som der står i ANSI C++ standart (En standart indenfor C++, for hvordan en kode skal være skrevet og opstillet. Fx kan man nu skrive <iostream> i stedet for <iostream.h> (.h = headerfile (som jeg vil komme ind på senere))) skal alle klasser, objekter og funktioner være defineret i std;. Istedet for at skrive using namespace std;
kunne vi også skrive std::cout << "hello word!";

int main() - int main() er en den funktion hvor dit program starter selvom main funktionen er i midten af dit program er det altid her programmet starter. Efter main kommer de to paranteser der betyder at main er en funtion, efter alle funktioner kommer der to paranteser.

{ - her starter din funktion.

cout << "Hello World!"; - cout står for console output og udskriver til dit standart output, som normalt er skærmen. Cout er deklereret i iostream headerfil, så hvis vi vil bruge cout skal vi altid inkludere iostream i vores program. Efter hver sætning skal der være et semikolon

std::cin.get(); - denne kommando sørger for at programmet holder "pause" indtil man trykker på en tast.

return 0; - her slutter dit program. return 0; er den mest almindelige måde at slutte programmet på. 0 er den værdi programmet returnerer.

} - her afslutter vi funktionen main.

Vores program kunne også have set sådan ud:

```
int main () { cout << " Hello World "; system("pause"); return 0; }
```

Det er bare for at gøre det lettere at læse, at man stiller det op på den måde. Det er en af de smarte ting ved Dev-C++ at den gør det automatisk.

En anden måde at kommentere sit program på er at bruge */* *kommentar* */*

som her:

```
/* Her kan du kommentere på flere
linjer */

#include <iostream>

using namespace std;


int main ()

{
  cout << "Hello World!";      // Her kan du kommentere på en linje
  std::cin.get();
  return 0;
}
```

Ny linje

Der er to måder at lave et linjeskrift på:

1. \n
2. endl;

\n kan benyttes sådan:

```
cout << "Forste linje \nAnden linje";
```

endl; kan benyttes sådan:

```
cout << "Forste linje" << endl;
cout << "Anden linje";
```

En anden sjov 'escape code' er \a som giver en bip lyd.
Prøv selv !

Artikel 2

Videre i c++

I denne artikel vil jeg gå lidt mere i dybten med c++. Jeg vil gennemgå flg. ting:

- Input (cin)
- Variabler
- Konstanter
- Udtryk
- Operatorer

Cin (console input)
Cin er en kommando fra C++'es standart input/output bibliotek, lige som cout. Den kan læse en indtastet værdi og gemme den i en variabel (som jeg kommer ind på senere). I dette eksempel bruger jeg cout, cin og en variabel:

```
#include <iostream>

using namespace std;

int main()
{
   int tal;                    // Her erklæres en variabel son i dette tilfælde er en Integer
   cout << "Indtast et tal: ";
   cin >> tal;                 // Her læser den det indtastede tal og gemmer den i variablen 'tal'
   cout << tal << endl;  // Her udskrives variablen 'tal'
   std::cin.get();
   return 0;
}
```

Har du evt brug for at få genopfrisket alle de andre sætninger som cout og using namespace std; kan du læse min første artikel om c++ her: http://www.eksperten.dk/artikler/762.

Lad os kigge nærmere på cin
Hvis du kigger godt på sætningen "cin >> tal;", vil du lægge mærke til at >> er istedet for << og det er fordi cin er et input og cout er et output.

Variabler

En variabel er en mængde plads, på computerens elektroniske hukomelse (bedre kendt som RAM eller Random Access Memory), som programmet 'reserverer' til at gemme en værdi (fx tallet 5).

Du kan selv bestemme værdien af en variabel ved at tildele den en værdi. Til det bruger du tildelingsoperatoren (=), som jeg vil komme ind på senere i denne artikel.
Det er faktisk meget simpelt at erklære en variabel. Her er et eksempel:

```
int a;
a = 22;
```

Her erklæres en variabel og den tildeles en værdien 22.
Dette kan også gøres i én sætning:

```
int a = 22;
```

Du kan selv bestemme navnet på din variabel, med nogle få untagelser som disse nøgle-ord i ANSI C++ standart:

```
asm, auto, bool, break, case, catch, char, class, const, const_cast, continue, default, delete, do, double, dynamic_cast, else, enum, explicit, extern, false, float, for, friend, goto, if, inline, int, long, mutable, namespace, new, operator, private, protected, public, register, reinterpret_cast, return, short, signed, sizeof, static, static_cast, struct, switch, template, this, throw, true, try, typedef, typeid, typename, union, unsigned, using, virtual, void, volatile, wchar_t
```

```
- Det er heller ikke klogt at bruge æøå I dine varibler da der ikke er alle compilere der opfatter disse som alm. bogstaver
```

Der findes forskellige typer af variabler, og de kaldes Data typer. Her er de så:

Da det er lidt svært at sætte det op her på exp.dk har jeg sat dem ind her:
www.faarup.1go.dk/typer.html

Jeg har medvilje ikke skrevet hvor meget data typerne fylder da det afhænger af den compiler du bruger, og din computer. Men det er der heldigvis råd for. I dette eksempel vil jeg vise hvordan man finder størrelsen af en data type, ved hjælp af funktionen sizeof() (som leveres af din compiler):

```
#include <iostream>

using namespace std;

int main()
{
   cout << "Storrelsen paa en double er: ";
   cout << sizeof(double) << " bytes." << endl;
   std::cin.get();
   return 0;
}
```

Mit program udskrev:
Storrelsen paa en double er: 8 bytes.

Dvs. at størrelsen (pladsen som variablen optager i computerens elektroniske hukommelse (RAM)) er på 8 bytes !
- læs evt mere om bytes her:
http://computer.howstuffworks.com/bytes.htm
eller på en af de mange andre sider:
http://www.google.dk/search?hl=da&q=bits+and+bytes&btnG=Google-s%C3%B8gning&meta=

- Prøv evt selv at finde resten af dine værdier på data typerne.

Konstanter

Konstanter minder meget om variabler, men der er dog den forskel at værdien på en konstant ikke kan ændres, efter man har erklæret den. Men kan fx godt først give en varibel værdien 1 og så senere i programmet ændre værdien til 2 eller til noget helt tredje !

Når du først har initialiseret (erklæret og tildelt en værdi) en konstant kan værdien ikke ændres senere i programmet.

En konstant erklæres men nøgleordet *const*

[/div]

const
Når du konstaterer en variabel med nøgleordet const kan du ikke ændre på værdien mere. Den bruges sådan:

```
const double PI = 3.1415;
```

Her kan man også bestemme typen af konstanten (data typen, her en double).

Der er også noget man kalder for opregnede konstanter men det vil jeg ikke komme ind på i denne artikel (måske i næste ;)).

Udtryk

Et udtryk er når man fx skriver x = a + b. Dette udtryk betyder ikke at x er lig a + b, men at x skal tildeles værdien af summen af a + b. Efter alle udtryk/sætninger skal der være et semikolon (;). Mellemrummene mellem x og = og mellem = og a osv. er noget som compileren ignorerer, så den kan godt undlades: x=a+b.

Til alle udtryk bruger man operatorer. Operatorer er et symbol der får compileren til at foretage en handling som fx at lægge to tal sammen.

Her er nogle brugbare (matematiske) operatorer:

```
+  = plus. Lægger to tal sammen
-   = minus. Trækker to tal fra hinanden
*  = gange. Ganger to tal
/   = dele. Deler to tal
%  = module. Finder resten af to delte tal.
```

Tildelingsoperatoren er =.
Tildelingsoperatoren kopierer data fra højre til venstre side.

fx

```
a = 1;
```

her kopieres værdien 1 som tildeles til a

! Pas på du ikke kommer til at blande den sammen med denne operator ==, som sammenligner data fra højre og venstre side.

Her er et lille eksempel på brug af dem begge:

```
if((a=b)==1)
```

her kopieres fra b til a
og sammenlign a med 1

...og et til lille eksempel:

```
a = (b == c);
```

her sammenlignes b og c og resultatet gemmes i a (forudsætter at a er af typen bool).

Da operatorer er et ret stort emne vil jeg ikke komme mere ind på det i denne artikel. Men jeg håber at du har fået fat i hvad det er.

Til sidst vil jeg angive et eksempel på hvordan man beregner arealet af en cirkel ud fra nogle indtastede værdier. Jeg gennemgår også alt det du har lært i denne artikel:

```
#include <iostream>

using namespace std;

main()
{
    double r, A;  // Her erklæres to doubler
    double pi = 3.1415;  // Her defineres en konstant
    cout << "Dette program beregner arealet af en cirkel" << endl;
    cout << "Indtast radius i cm: ";
    cin >> r;  // Her læser den det indtastede tal og gemmer den i variablen 'r'
    A = pi * r * r;  // Her foretages en beregning (et udtryk)
    cout << "Arealet af cirklen er: " << A << " cm2." << endl;  // Her udskrives resultatet
    system("pause");
    return 0;
}
```

Dette program beregner arealet af en cirkel
Indtast radius i cm: *4*
Arealet af cirklen er: 50.264 cm2.

Slut på artikel 1
Det var så slut på artikel 1 om c++ for begyndere. Jeg håber du har fået et godt indblik i c++ og måske har fået lidt interesse for det. I undrer jeg sikkert over at jeg har skrevet slut på artikel 1, og det er der en grund til. Jeg arbejder allerede nu på en 2'er som vil gå lidt mere i dybten med c++.

Alt kritik vil blive modtaget, så hvis du kan finde en stavefejl eller en anden mangel i artiklen så skriv endelig.

Hvis du ikke kan vente med at lære mere c++ så vil jeg anbefale at låne eller købe bogen
C++ Grundbog af Jesse Liberty
som kan findes på: http://idgforlag.dk/
Eller nogle af de mange tutorials der findes (mest på engelsk) på nettet.

Masser af simple eksempler på www.faarup.1go.dk

> Jeg kan ikke anslå en dato for hvornår c++ for begyndere 3 kommer men det bliver nok i starten af oktober.

De værste fejl skulle være rettet nu. Ellers skriv til mig.

MVH VISUALDEVELOPER

Kommentar af borrisholt d. 18. Aug 2005 | 1

#define er ikke en konstant, det er direkte forkert at skrive det !

Kommentar af qtax87 (nedlagt brugerprofil) d. 12. Oct 2006 | 2

Hos mig melder den fejl ved det 1 først eksemple.

Kommentar af bertelbrander d. 17. Aug 2005 | 3

Hvis de værste fejl blev fjernet kunne det blive en god artikel.
Se også:
http://www.eksperten.dk/spm/639888

Kommentar af lamaduck d. 27. Sep 2005 | 4

Kommentar af madsgr d. 19. Aug 2005 | 5

God artikel til begyndere der vil lid mere end bare se hvad det er !!

Kommentar af iostream d. 17. Aug 2005 | 6

Endnu bedre end nr. 1

Kommentar af peter_bf d. 25. Oct 2005 | 7

hov nu kom jeg til også at købe denne artikel (ikke så smart at have 2 artikler)
ellers god artikel, se evt min kommentar i den anden !

Kommentar af nothingaqua d. 25. Mar 2006 | 8

sidste eksempel
double pi = 3.1415; // Her defineres en konstant
skal vel være
const double pi = 3.1415; // Her defineres en konstant ?

Kommentar af mattiasdh d. 01. May 2006 | 9

??

Kan man bruge dette i "AUTOIT"??

Kan ikke få NOGET af dette til at virke..

??