

```
>>> C-Programmering for begyndere  
>>> Del 1 - Hello World, Datatyper, Matematik og I/O
```

Name: Jacob B. Pedersen<sup>†</sup> og Jakob S. Nielsen<sup>‡</sup>

Date: March 19, 2018

---

<sup>†</sup>jacob.bp@mvb.net

<sup>‡</sup>indsæt mail her

>>> Indhold

1. Jeres undervisere

Hvem er vi?

2. Intro til C

Hvad er C?

C's historie

Eksempler på brug af C

3. Et C-programs opbygning

Et C-programs opbygning

4. Øvelseseksempler

Opsætning af CodeLite

Eksempel 1 - Hello World

Eksempel 2 - Datatyper

Eksempel 3 - Matematik med Datatyper

Eksempel 4 - Input og Output

5. Kreative Opgaver

>>> Hvem er vi?



- \* Jacob Bechmann Pedersen
  - \* Læser Elektronikingeniør
  - \* 6. Semester
- \* Direkte fra HTX
- \* Startet egen virksomhed
  - \* Bechmann & Vang - Med fokus på elektroniske musikinstrumenter
- \* Holder også Arduino workshops



- \* Jakob Skjødt Nielsen
  - \* Læser Elektronikingeniør
  - \* 2. Semester
- \* Elektronikfagtekniker ved B&O
- \* Osv.
- \* Osv.

>>> Hvad er C?

- \* C er letvægtigt og hurtigt!
  - \* Fylder meget lidt plads
  - \* Kører meget stærkt!
- \* Kompatibelt og portabelt til de fleste systemer
- \* Kan bruges bredt!
  - \* Kan gå dybt i detaljerne
  - \* Eller behandle i overfalden
- \* C er et compiled sprog
  - \* Koden samles til 1'er og 0'er én gang
  - \* Inden det køres, modsat interpreted

# THE C PROGRAMMING LANGUAGE

## >>> C's historie

- \* C begyndte som så meget andet i Bell Labs
  - \* Skabt af Dennis Ritchie
  - \* Baseret på B af Ken Thompson
  - \* Udviklet til arbejdet på UNIX
- \* Udviklet af programmører til programmører!



## >>> Eksempler på brug af C

- \* Kernen til dit Operativsystem er hovedsagligt skrevet i C
- \* C har inspireret mange sprog
  - \* Swift
  - \* Java
  - \* PHP
  - \* R
- \* Compilers og interpreters er ofte skrevet i C!
- \* C bruges i indlejret elektronik i stor stil!

### Answers



We use almost entirely C, C++, and C# for Windows.

ryan



Ryan Waite - Product Unit Manager - Windows HPC

4



Swift



Java



>>> Et C-programs opbygning

\* Linje 3:

\* main() er  
hovedfunktionen

\* Linje 6:

\* Funktioner "returnerer"  
ofte status til sidst

\* Linje 1:

\* Libraries med  
standardkomponenter

\* Linje 5:

\* printf() er en af dem.  
Den skriver til konsollen

Output:

---

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      printf("Hello
              world\n");
6      return 0;
7  }
```

---

---

Hello world

---

## >>> Et C-programs opbygning

\* Linje 3:

\* Man kan deklarere  
variable

\* Linje 10:

\* Og lægge dem sammen

\* I en ny variabel

\* Linje 11:

\* Skrive dem ud,  
formateret

---

```
1  #include <stdio.h>
2
3  int a;
4  int b;
5
6  int main(void)
7  {
8      a = 2;
9      b = 3;
10     int c = a + b;
11     printf("%d + %d =
        %d\n", a, b, c);
12     return 0;
13 }
```

---

Output:

---

2 + 3 = 5

---



>>> Opsætning af CodeLite

- \* Nu skal vi have opsat CodeLite
  - \* <https://downloads.codelite.org/>
- \* Hvad man kalder et IDE
  - \* Integrated Development Environment
- \* Det gør vi live!

>>> Eksempel 1 - Hello World

- \* Det første program hedder Hello World
- \* Det er en klassiker inden for programmering
  - \* Første program man laver, så der er hul igennem
- \* `../Del_1/Examples/1_Hello_World/main.c`
- \* Vi tager den i fællesskab!

>>> Eksempel 1 - Hello World

```
1  #include <stdio.h>
2
3  int main(int argc, char **argv)
4  {
5      printf("hello world\n");
6      return 0;
7  }
```

Output:

hello world

## >>> Eksempel 2 - Datatyper

- \* Vi skal lære forskellen på datatyper i C
- \* Der er en række forskellige:
  - \* int - heltal/integer
  - \* floating - floating point kommatall
  - \* char - skrifttegn/character
  - \* bool - sand/falsk (implementeret i stdbool.h)
- \* `../Del_1/Examples/2_Datatypes/main.c`
- \* Vi tager den i fællesskab!

## >>> Eksempel 2 - Datatyper

---

```
1  #include <stdio.h>
2
3  int main(int argc, char **argv)
4  {
5      int a = 10; //integer
6      float b = 2.5; //float
7
8      printf("an integer is a whole number. integer 'a' is:
9              %d\n\n",a);
10     printf("floats can store decimals as well. float 'b' is :
11             %f\n\n",b);
12     printf("we don't always want that many decimals though! you
13             can adjust that like this %.2f\n\n",b);
14
15     char ch = 'A'; //char
16     printf("a character has several values, the character value
17             is %c, and the integer value is %d\n\n",ch,ch);
18     return 0;
19 }
```

## >>> Eksempel 2 - Datatyper

Output:

---

an integer is a whole number. integer 'a' is: 10

floats can store decimals as well. float 'b' is : 2.500000

we don't always want that many decimals though! you can adjust  
that like this 2.50

a character has several values, the character value is A, and  
the integer value is 65

---

## >>> Eksempel 3 - Matematik med Datatyper

- \* Hvordan regnes med de forskellige datatyper?
- \* Vi ser lidt på basal aritmetik
  - \* Multiplikation
  - \* Division
- \* Og hvordan variablene reagerer
- \* `../Del_1/Examples/3_Datatype_Math/main.c`
- \* Vi tager den i fællesskab!

## >>> Eksempel 3 - Matematik med Datatyper

```
1  #include <stdio.h>
2
3  int main(int argc, char **argv)
4  {
5      int a = 10;
6      int b = 3;
7
8      printf("A quick way to do maths: %d * %d is %d\n\n", a,b,a*b);
9
10     int c = a * b;
11
12     printf("displaying the math in variable c: %d * %d is %d\n\n",a,b,c);
13
14     c = a / b;
15
16     printf("wrong datatypes can cause some problems: %d / %d as an integer result: %d\n\n",a,b,c);
17
18     float d = 10;
19     float e = 3;
20     float f = d / e;
21
22     printf("Now with a float format specifier: %.1f / %.1f as a float result: %.2f\n\n",d,e,f);
23
24     return 0;
25 }
```



## >>> Eksempel 3 - Matematik med Datatyper

### Output:

---

A quick way to do maths:  $10 * 3$  is 30

displaying the math in variable c:  $10 * 3$  is 30

wrong datatypes can cause some problems:  $10 / 3$  as an integer  
result: 3

Now with a float format specifier:  $10.0 / 3.0$  as a float  
result: 3.33

---

## >>> Eksempel 4 - Input og Output

- \* Vores programmer har været meget ensidige
  - \* Data har været "hardcoded"
  - \* Vi har brug for input!
- \* scanf() kan afhjælpe dette!
  - \* Hører til i samme pakke som printf
  - \* stdio.h
- \* ../Del\_1/Examples/4\_Input\_Output/main.c
- \* Vi tager den i fællesskab!

## >>> Eksempel 4 - Input og Output

```
1  #include <stdio.h>
2
3  int main(int argc, char **argv)
4  {
5      float input;
6
7      printf("Please input a number: "); //print input prompt
8      scanf("%f",&input); //take float value as input
9
10     printf("\nHere is your entered number: %.2f\n\n",input);
11
12     float input1;
13     float input2;
14     printf("enter two numbers to be multiplied, hit enter after each number:\n");
15     scanf("%f %f",&input1,&input2);
16
17     float result = input1 * input2;
18     printf("result: %.2f * %.2f = %.2f\n\n",input1,input2,result);
19
20     return 0;
21 }
```

```
>>> Eksempel 4 - Input og Output
```

Output:

---

```
Please input a number: 3
```

```
Here is your entered number: 3.00
```

```
enter two numbers to be multiplied, hit enter after each number:
```

```
3
```

```
3
```

```
result: 3.00 * 3.00 = 9.00
```

---

## >>> Kreative Opgaver

- \* Der ligger kreative opgaver tilgængelige:
  - \* [../Del\\_1/Exercises/C\\_exercises\\_1\\_dansk.pdf](#)
- \* Der er hjælp at hente her på workshopen
- \* God arbejdslyst! - Happy Hacking!