



CPROUC/2021-2022

Jesse op den Brouw

CPROUC

Introductie

DE HAAGSE
HOGESCHOOL

CPROUC

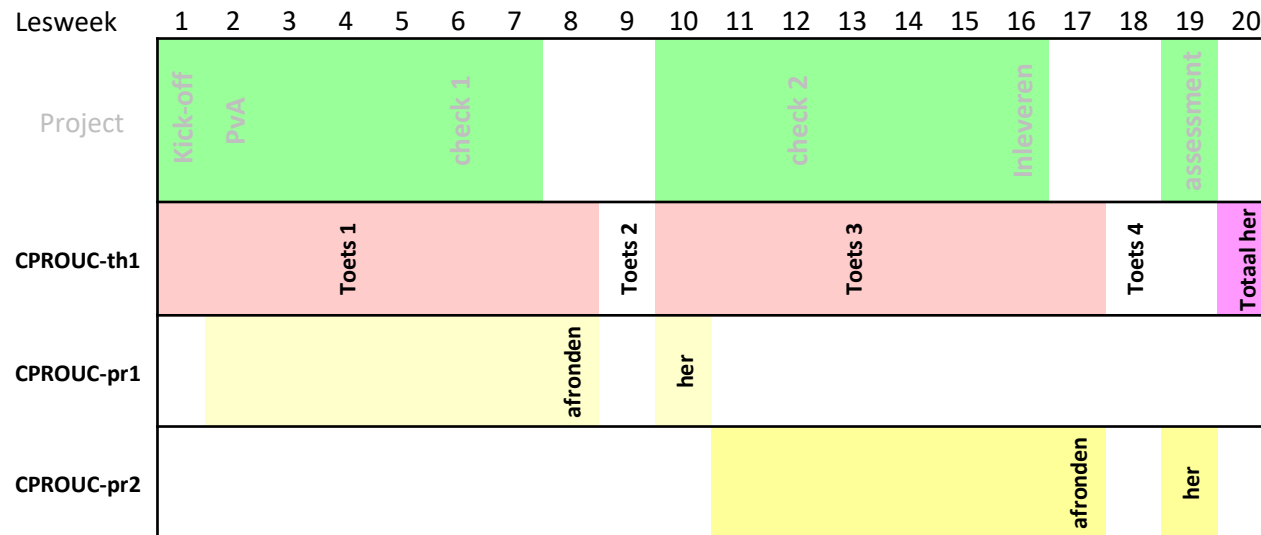
- CPROUC = Programmeren in C en basis microcontrollertechniek
- Programmeren met behulp van de taal C op een PC en een Arduino
- Het vak duurt 20 weken (semestervak)
- Het vak bestaat uit theorie, practicum en toetsing
- Docenten zijn Jesse op den Brouw, Ad van den Bergh en Willem-Pieter Zoutendijk (alleen practicum)
- Email: J.E.J.opdenBrouw@hhs.nl

Plaats in het curriculum van CPROUC

- Voorbereiding voor:
 - Projecten in het vervolg van de opleiding
 - UCPROG (Microcontroller programmeren)
 - OOSPLC (Object georiënteerd programmeren in C++)
 - Minor Embedded Systems



Overzicht lesweken



- Projectenlijn is niet van toepassing
- Theorie begint in lesweek 1 en lesweek 10
- Practicum begint in lesweek 2 en lesweek 11

Werkvormen theorie

- CPROUC-th1
 - 2 lesuren theorie per week (totaal 16x2 lesuren)
 - Docenten zijn Jesse op den Brouw en Ad van den Bergh
- Behandelen van de lesstof a.d.h.v. de voorgeschreven literatuur.
- Beantwoorden van vragen (stel ze!)
- Uitwerken opgaven
- Colleges starten in lesweek 1 en lesweek 10

Lesweek	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Project	Project								Project								Project				
CPROUC-th1	Toets 1								Toets 2	Toets 3								Toets 4	Totaal	her	
CPROUC-pr1	afro							en	her												
CPROUC-pr2											afro							en	her		

Werkvormen practicum

- CPROUC-pr1/pr2
 - 2 lesuren practicum per week (totaal 14x2 lesuren)
 - Docenten zijn Jesse op den Brouw, Ad van den Bergh en Willem-Pieter Zoutendijk
- Uitwerken van de opgegeven opdrachten
- Tijdens de practica worden opdrachten afgetekend
- Gebruik een laptop
- Practica starten in lesweek 2 en lesweek 11
- Aanwezigheid is verplicht!

Lesweek	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Project	Project								Project								Project				
CPROUC-th1	Toets 1								Toets 2	Toets 3								Toets 4	Totaal	her	
CPROUC-pr1	afro							en	her												
CPROUC-pr2											afro							en	her		

Toetsing

- Toetsing theorie
 - In lesweek 4, 9, 13, 18, 20 (her)
 - Toetsing is cumulatief over de stof (10, 20, 30, 40 punten)
 - Hertoets is over de gehele stof (cumulatieve toetsen vervallen)
 - Gesloten boek, dus geen hulpmiddelen
- Toetsing practicum
 - In lesweek 8, 10 (her), 17, 19 (her)

Lesweek	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Project	Project								Project							Project					
CPROUC-th1	Toets 1								Toets 2	Toets 3						Toets 4	Totaal	her			
CPROUC-pr1	afro							en	her												
CPROUC-pr2											afro						en	her			

Leermiddelen

- Boek: Jesse op den Brouw, *De programmeertaal C*
 - Dit boek is te vinden via BlackBoard
 - Broncode is te vinden via https://github.com/jesseopdenbrouw/book_c
- Dictaat: Ad van den Bergh, *Creatief met Arduino*
 - Dit dictaat is binnenkort te vinden op BlackBoard

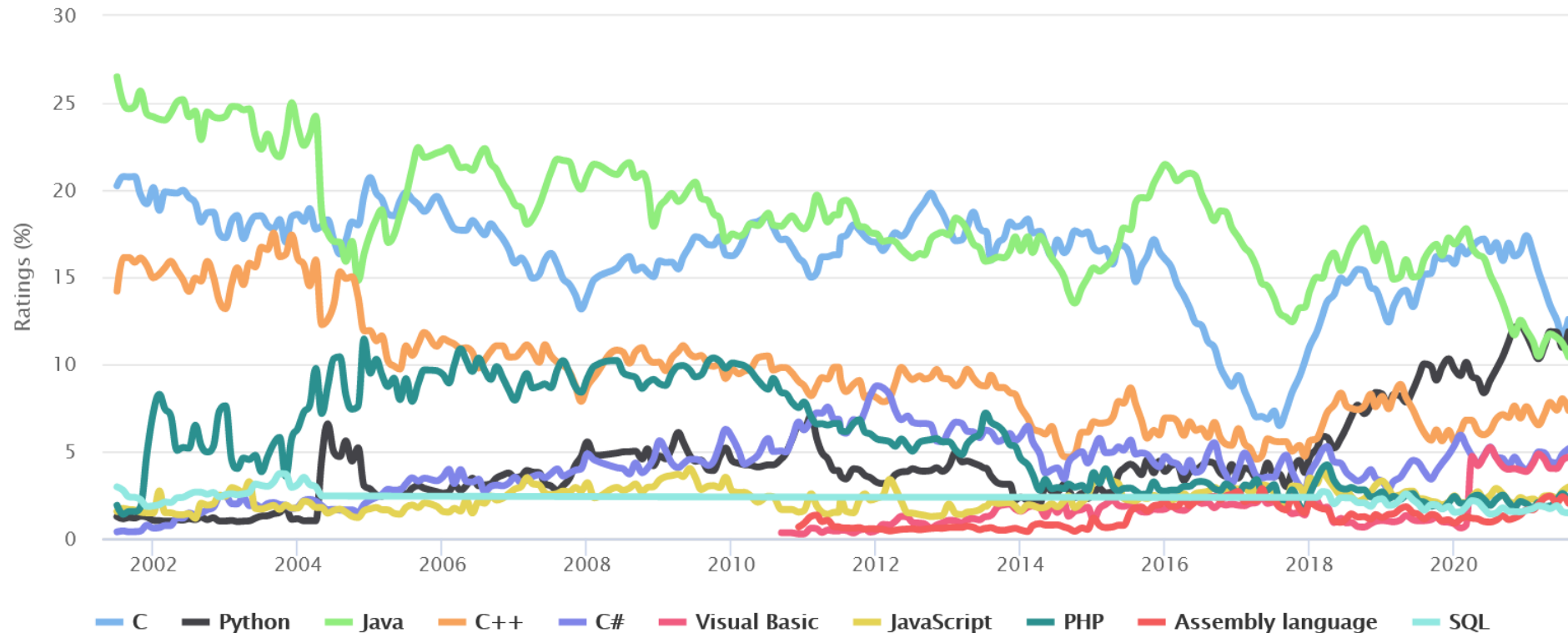
Leermiddelen

- Gebruik van een ontwikkelomgeving C programmeren
 - Code::Blocks 20.03 of Visual Studio 2019 Community Edition
 - Mac OS-X: Xcode
 - Alle voorbeelden zijn uitgewerkt met Visual Studio 2019 Community
- Gebruik van een ontwikkelomgeving voor de Arduino
 - Arduino IDE

Wat weet je al?

- Welke programmeertalen ken jij al?
 - C, Python, C#, Java, ...
- Waarom met de programmeertaal C?
 - Wordt veel gebruikt op microcontrollers
- Waarom leren programmeren bij Elektrotechniek?
 - Veel moderne elektronische systemen zijn uitgerust met een kleine computer

Waarom programmeren in C?



<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Wat is programmeren?

- Een programma vertelt een computer wat die moet doen.
- Welke basisbewerkingen zijn er?
 - Rekenen
 - Herhalen
 - Beslissen
- Delegeren (verdeel en heers, functies)
- Structureren van data (array en structure)
- Complexe bewerkingen (met behulp van pointers)

Gestructureerd programmeren

- Wat is gestructureerd programmeren?
- Een (programmeer)-paradigma
 - Een stelsel van modellen en theorieën dat, binnen een gegeven wetenschappelijke discipline, het denkkader vormt van waaruit de werkelijkheid geanalyseerd en beschreven wordt.
- Subdiscipline van procedureel programmeren (imperatief).
- Wat is procedureel programmeren?
 - Een programma wordt opgesteld in de vorm van opdrachten of statements die direct kunnen worden uitgevoerd en de state (toestand) van het programma aanpassen.

Inhoud CPROUC

- Gestructureerd Programmeren in C
 - Invoer en uitvoer (printf en scanf)
 - Rekenen met gehele (int) en floating point (double) getallen.
 - Herhalingsopdrachten (while, do while en for)
 - Beslissingsopdrachten (if, if else en switch case)
 - Functies
 - Arrays
 - Karakters en strings
 - Structures (struct)

Inhoud CPROUC

- Gestructureerd Programmeren in C (vervolg)
 - Pointers
 - Tekst files
 - C preprocessor
 - Pointers naar functies
- Programmeren in C op de Arduino Uno R3
 - Lezen van en schrijven naar IO-poorten
 - Digitaal en analoge waarden inlezen
 - Gebruik van polling en interrupts
 - Schuifregisters en multiplexing
 - Timing en toetsafhandeling

Minimaal C-programma

- Een C-programma moet altijd de functie main bevatten.

```
// Commentaarregel
/* Blokcommentaar mag over
   meerdere regels */
int main(void)
{
    return 0;
}
```

- De functie main geeft een waarde terug aan het operating system (Windows, Mac OS-X, Linux) bij afsluiten.

Afdrukken op het scherm

- Met de functie `printf` kan je tekst en variabelen op het scherm afdrukken.

```
#include <stdio.h>           // Header file

int main(void)              // De functie main
{
    int a;                  // Definitie integer
    a = 6;                  // Toekenning

    printf("De waarde van a is: %d\n", a);
    return 0;              // Terugkeer
}
```

Afdrukken op het scherm

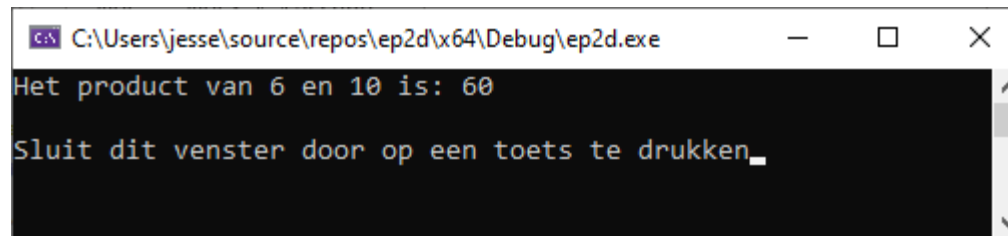
```
#include <stdio.h>

int main(void) {
    int a, b, product;    // Definitie variabelen
    a = 6;
    b = 10;
    product = a * b;     // Toekenning met bewerking

    printf("Het product van %d en %d is: %d\n", a, b, product);
    printf("\nSluit dit venster door op een toets te drukken");

    getchar();          // Wacht tot druk op enter-toets

    return 0;
}
```



```
C:\Users\jesse\source\repos\ep2d\x64\Debug\ep2d.exe
Het product van 6 en 10 is: 60
Sluit dit venster door op een toets te drukken_
```



CPROUC/2021-2022

Jesse op den Brouw

CPROUC

Variabelen

DE HAAGSE
HOGESCHOOL

Variabelen

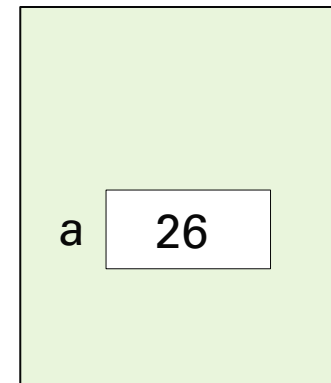
- Een variabele is een klein stukje opslagruimte in het werkgeheugen van de computer. Voorbeeld:

```
int a;
```

- In het geheugen is er nu een stukje ruimte vrijgemaakt en deze heeft de naam a.
- Met behulp van een toekenning kun je waarden stoppen in deze variabele:

```
a = 26;
```

Computergeheugen



Basis variabelen

- Een variabele heeft een bepaald type. Dit vertelt hoeveel opslagruimte in het geheugen nodig is en hoe deze opslagruimte gebruikt wordt.
- Voorbeelden van type variabelen (gehele unsigned variabelen):

unsigned char	8 bits	0 – 255
unsigned int	32 bits	0 – 4.294.967.295
- Een eenheid van 8 bits wordt een *byte* genoemd.
- Een eenheid van 32 bits (4 bytes) wordt een *word* genoemd.

Basis variabelen

- Gehele signed variabelen:

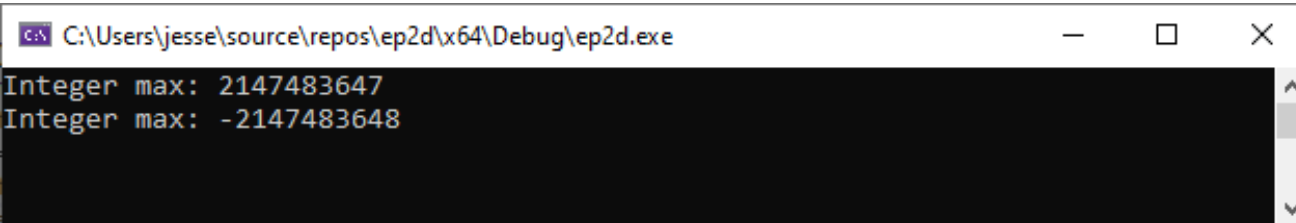
signed char	8 bits	-128 – +127
signed int	32 bits	-2.147.483.648 – +2.147.483.647

- Let op: een (unsigned/signed) int is bij de meeste grote processoren 32 bits maar bij sommige (microcontrollers) 16 bits (bijvoorbeeld Arduino Uno).

Basis variabelen

- Signed integers worden in two's complement opgeslagen.
- Er is geen test op overflow.

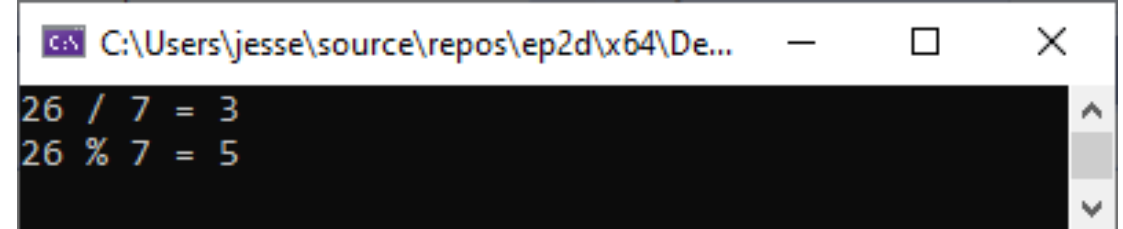
```
int main() {  
    int smax = 2147483647;  
    printf("Integer max: %d\n", smax);  
    smax = smax + 1;  
    printf("Integer max: %d\n", smax);  
    return 0;  
}
```



```
C:\Users\jesse\source\repos\ep2d\x64\Debug\ep2d.exe  
Integer max: 2147483647  
Integer max: -2147483648
```


Rekenen met gehele getallen

- Vijf basisbewerkingen
 - Optellen: $a + b$
 - Aftrekken: $a - b$
 - Vermenigvuldigen: $a * b$
 - Delen: a / b
 - Modulo: $a \% b$

A screenshot of a terminal window with a black background and white text. The window title bar shows the path 'C:\Users\jesse\source\repos\ep2d\x64\De...'. The terminal displays two lines of output: '26 / 7 = 3' and '26 % 7 = 5'.

```
C:\Users\jesse\source\repos\ep2d\x64\De...  
26 / 7 = 3  
26 % 7 = 5
```

```
int a = 26, b = 7;  
printf("%d / %d = %d\n", a, b, a / b);  
printf("%d %% %d = %d\n", a, b, a % b);
```

Rekenen met gehele getallen

- Let op met delingen, er wordt naar beneden (naar de 0 toe) afgerond

```
#include <stdio.h>
```

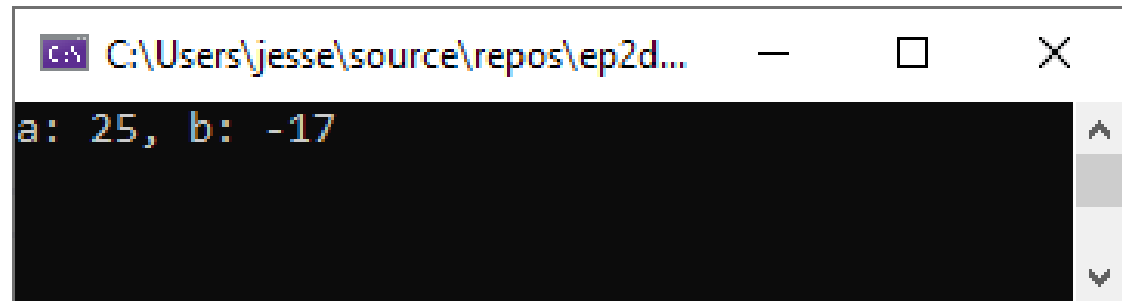
```
int main(void) {
```

```
    int a = 25.7;
```

```
    int b = -17.7;
```

```
    printf("a: %d, b: %d\n", a, b);
```

```
}
```



```
C:\Users\jesse\source\repos\ep2d...  
a: 25, b: -17
```

Rekenen met gehele getallen

- Let op de volgorde van de bewerkingen:

```
#include <stdio.h>
```

```
int main(void) {
```

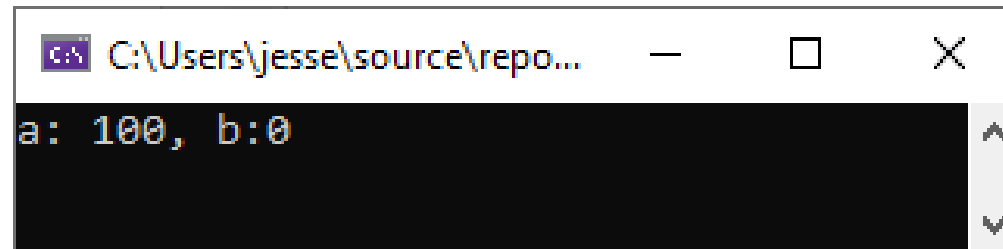
```
    int a = 100;
```

```
    int b ;
```

```
    b = 5 / 9 * a;
```

```
    printf("a: %d, b:%d\n", a, b);
```

```
}
```

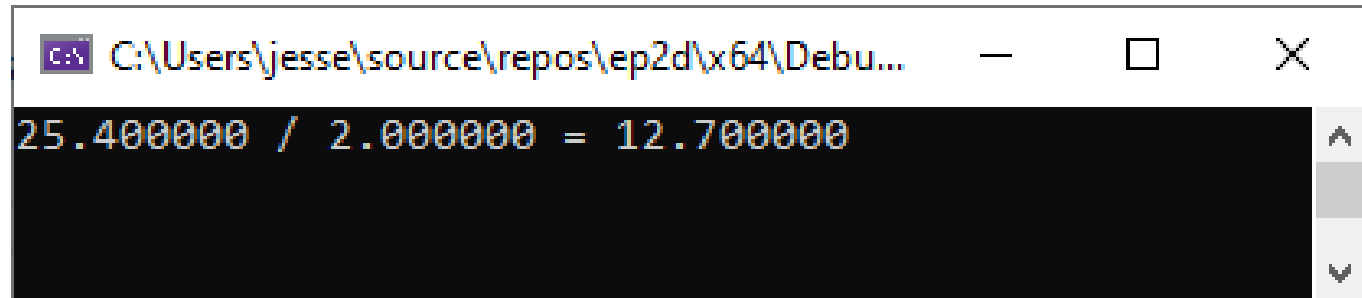


```
C:\Users\jesse\source\repo...  
a: 100, b:0
```

Rekenen met floating point getallen

- Vier basisbewerkingen
 - Optellen: $a + b$
 - Aftrekken: $a - b$
 - Vermenigvuldigen: $a * b$
 - Delen: a / b

```
double a = 25.4, b = 2.0;  
printf("%f / %f = %f\n", a, b, a / b);
```



```
C:\Users\jesse\source\repos\ep2d\x64\Debu...  
25.400000 / 2.000000 = 12.700000
```

Rekenen met floating point getallen

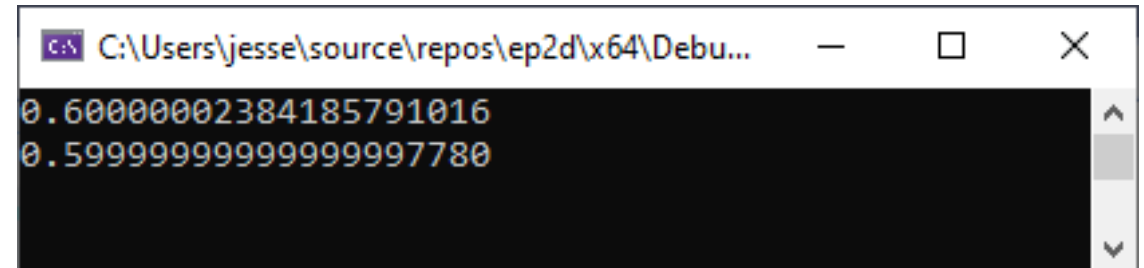
- Een float wordt opgeslagen in 32 bits en is niet erg nauwkeurig. Minimaal 6 (max 9) significante cijfers.
- Een double wordt opgeslagen in 64 bits en is ongeveer 2x zo nauwkeurig. Minimaal 15 (max 17) significante cijfers.
- Bij afdrucken wordt een float tijdelijk omgezet in een double.
- Bij inlezen niet!

Niet alles kan...

- Niet alle “mooie” decimale breuken kunnen exact worden weergegeven in floating point getallen.

```
#include <stdio.h>

int main(void) {
    float f_nulkommazes = 0.6f;
    double d_nulkommazes = 0.6;
    printf("%.20f\n", f_nulkommazes);
    printf("%.20f\n", d_nulkommazes);
    getchar();
    return 0;
}
```



```
C:\Users\jesse\source\repos\ep2d\x64\Debu...
0.60000002384185791016
0.59999999999999997780
```

Variabelen inlezen

- Met de functie scanf kan een waarde worden ingelezen en aan een variabele worden toegekend.

```
int a;
```

```
float b;
```

```
double c;
```

```
scanf("%d", &a);    /* lees integer van toetsenbord */
```

```
scanf("%f", &b);    /* lees float van het toetsenbord */
```

```
scanf("%lf", &c);   /* lees double van het toetsenbord */
```

```
/* lees integer, float en double van het toetsenbord */
```

```
scanf("%d %f %lf", &a, &b, &c);
```

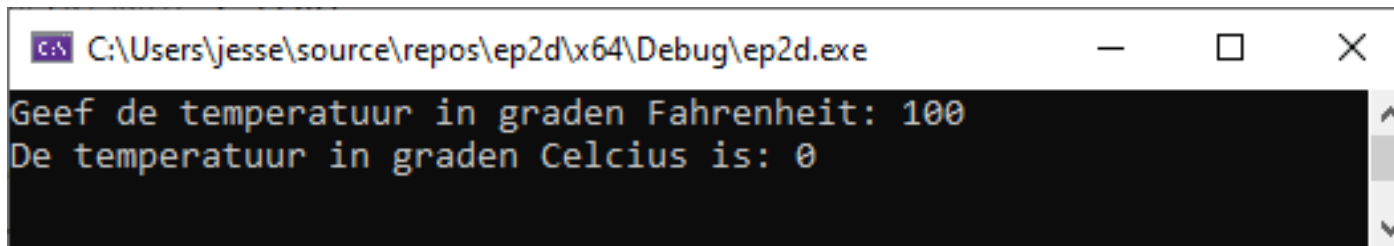
Voorbeeld

- Schrijf een programma waarmee een temperatuur in graden Fahrenheit omgerekend kan worden naar graden Celsius.
- De temperatuur in graden Fahrenheit is een geheel getal.
- De temperatuur in graden Celsius moet worden afgerond tot een geheel getal
- Omzetten met de bekende formule: $T_C = (T_F - 32) \cdot \frac{5}{9}$
- Voorbeeld: $T_F = 100 \rightarrow T_C = 37,777 \rightarrow T_C = 38$
- Voorbeeld: $T_F = 0 \rightarrow T_C = -17,777 \rightarrow T_C = -18$

Voorbeeld

```
#include <stdio.h>
```

```
int main(void) {  
    int C, F;  
    printf("Geef de temperatuur in graden Fahrenheit: ");  
    scanf("%d", &F);  
    C = (F - 32) * (5 / 9);  
    printf("De temperatuur in graden Celcius is: %d\n", C);  
    getchar();  
    return 0;  
}
```

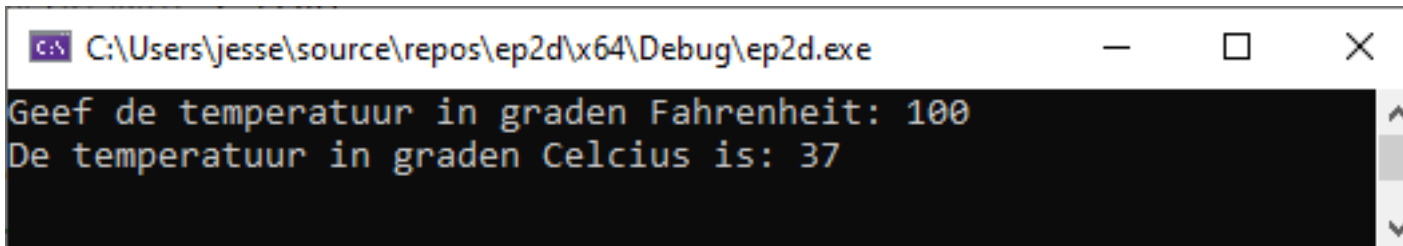


```
C:\Users\jesse\source\repos\ep2d\x64\Debug\ep2d.exe  
Geef de temperatuur in graden Fahrenheit: 100  
De temperatuur in graden Celcius is: 0
```

Voorbeeld

```
#include <stdio.h>
```

```
int main(void) {  
    int C, F;  
    printf("Geef de temperatuur in graden Fahrenheit: ");  
    scanf("%d", &F);  
    C = (F - 32) * 5 / 9;  
    printf("De temperatuur in graden Celcius is: %d\n", C);  
    getchar();  
    return 0;  
}
```

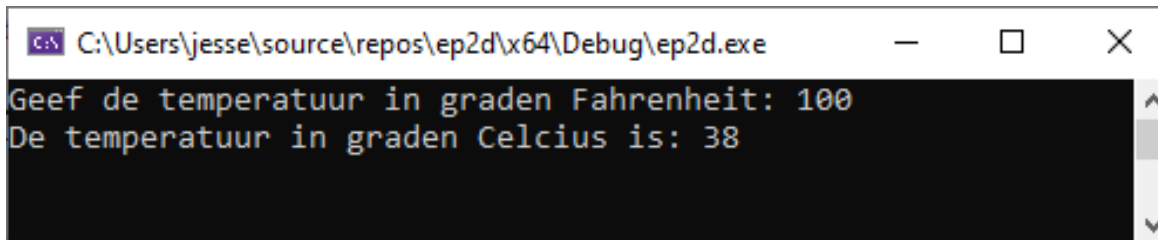


```
C:\Users\jesse\source\repos\ep2d\x64\Debug\ep2d.exe  
Geef de temperatuur in graden Fahrenheit: 100  
De temperatuur in graden Celcius is: 37
```

Voorbeeld

```
#include <stdio.h>
```

```
int main(void) {  
    int C, F;  
    printf("Geef de temperatuur in graden Fahrenheit: ");  
    scanf("%d", &F);  
    C = (F - 32) * 5.0 / 9 + 0.5;  
    printf("De temperatuur in graden Celcius is: %d\n", C);  
    getchar();  
    return 0;  
}
```

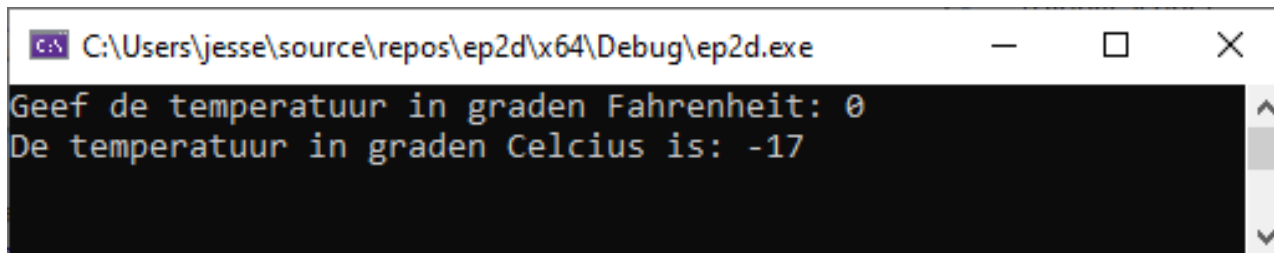


```
C:\Users\jesse\source\repos\ep2d\x64\Debug\ep2d.exe  
Geef de temperatuur in graden Fahrenheit: 100  
De temperatuur in graden Celcius is: 38
```

Voorbeeld

```
#include <stdio.h>
```

```
int main(void) {  
    int C, F;  
    printf("Geef de temperatuur in graden Fahrenheit: ");  
    scanf("%d", &F);  
    C = (F - 32) * 5.0 / 9 + 0.5;  
    printf("De temperatuur in graden Celcius is: %d\n", C);  
    getchar();  
    return 0;  
}
```



```
C:\Users\jesse\source\repos\ep2d\x64\Debug\ep2d.exe  
Geef de temperatuur in graden Fahrenheit: 0  
De temperatuur in graden Celcius is: -17
```

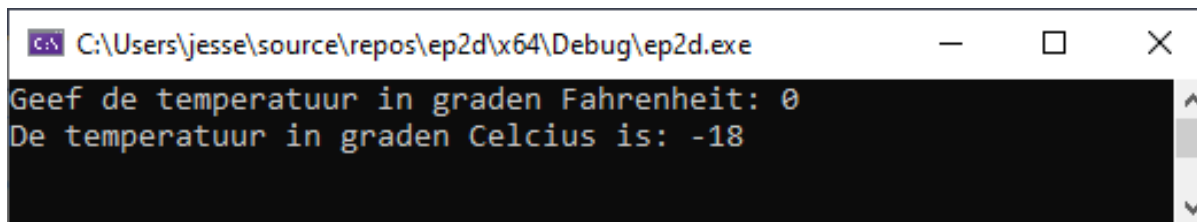
Voorbeeld

```
int main(void) {
    int C, F;
    double Cdouble;
    printf("Geef de temperatuur in graden Fahrenheit: ");
    scanf("%d", &F);
    Cdouble = (F - 32) * 5.0 / 9;
    if (Cdouble > 0) {
        C = Cdouble + 0.5;
    }
    else {
        C = Cdouble - 0.5;
    }
    printf("De temperatuur in graden Celcius is: %d\n", C);
    getchar();
    return 0;
}
```

Voorbeeld

```
#include <stdio.h>

int main(void) {
    int F;
    double C;
    printf("Geef de temperatuur in graden Fahrenheit: ");
    scanf("%d", &F);
    C = (F - 32) * 5.0 / 9;
    printf("De temperatuur in graden Celcius is: %.0f\n", C);
    getchar();
    return 0;
}
```



```
C:\Users\jesse\source\repos\ep2d\x64\Debug\ep2d.exe
Geef de temperatuur in graden Fahrenheit: 0
De temperatuur in graden Celcius is: -18
```

let's change