

C++

Programmeren

TIPR001-5B



Vereiste voorkennis

- Programmeren 1 (TINPRO02-1)
- Programmeren 2 (TINPRO02-2)
- Programmeren 3 (TINPRO02-3)
- Programmeren 4 (TINPRO02-3)

Cursusinformatie

- 7 weken + 1 uitloopweek
- 2 studiepunten
- Tentamen: eindopdracht met mondelinge toelichting

Programma

Week Lesinhoud

-
- | | |
|---|--|
| 1 | Basis C++ syntax: filestructuur, variabelen en constanten (basistypes en andere types), flow control statements, arithmetic, relational, logical en bitwise operatoren |
| 2 | Pointers, de address operator en de reference declaraties en hun toepassing in functies |
| 3 | Basis OO programmeren: Classes met methods en attributes declareren, objecten instantiëren. Eenvoudige constructors en destructors |
| 4 | Console en file I/O |
| 5 | Toelichting ingeleverde ontwerp van eindopdracht |
| 6 | OO programmeren: overerving |
| 7 | Praktijkoefeningen en eventueel bonus: Polymorfe datastructuren met toepassing van abstract base classes en virtual function overriding |
| 8 | Uitloopweek |
-

Leermiddelen

Verplicht:

- Online versie van het boek “The C++ programming language”, by Bjarne Stroustrup, 4th Edition, Addison Wesley te vinden op: <https://archive.org/details/2013TheCProgrammingLanguageBjarneStroustrup/page/n0>
- Arduino ontwikkelbord en programmeeromgeving,
- C++ ontwikkelomgeving

Facultatief:

- Programming - Principles and Practice Using C++ , by Bjarne Stroustrup, 2nd Edition, Pearson Education (hard copy)
 - C++ Primer, by Stanley Lippman and Josee Lajoie,
- Aan de slag met C++, Gertjan Laan van academicx

Werkwijze

- Twee klassen in parallel naast elkaar
- Verdeling, indien mogelijk, in basis en gevorderden klas
- Eerste verdeling o.b.v. test en eigen inzicht
- Doorstromen altijd mogelijk
- Lesblokken van 3 uur: theorie en praktijk wisselen elkaar om

Codeerstijl

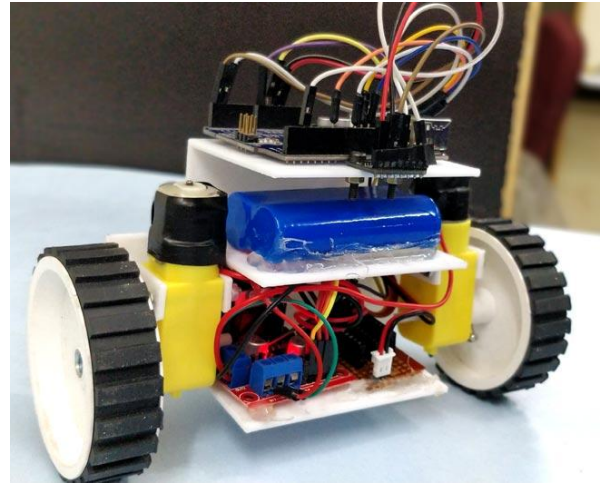
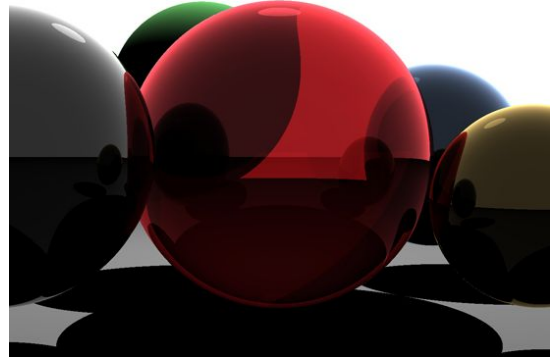
Zowel tijdens de cursus als tijdens het assessment wordt gekeken naar codeerstijl, en wordt de TI-Codeerstijl3x3 (zie classroom) gevolgd.

TI-CodeerStijl3x3 is een set regels die gevolgd worden tijdens het schrijven van broncode.

Eindopdrachten

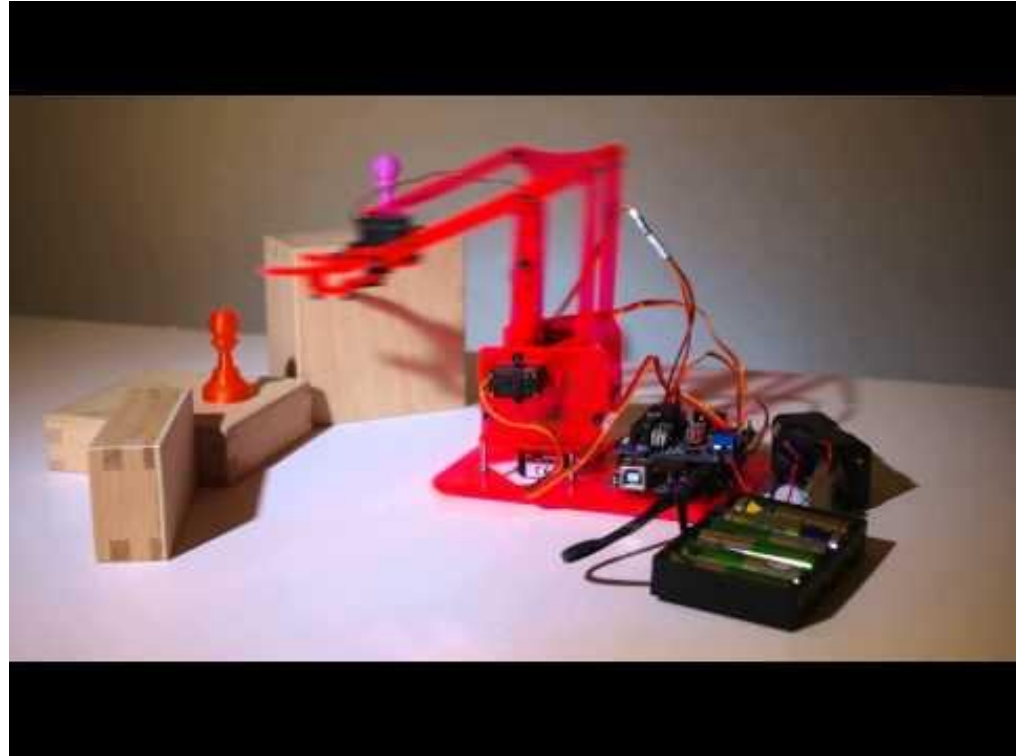
Vrije keuze uit twee opdrachten:

- Puur software opdracht: ray tracer
 - <https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-overview>
- Hardware+software opdracht: Zelfbalancerende robot
 - Zelfbouwpakket
 - Bouwtijd valt buiten de zelfstudietijd voor deze cursus
 - Delen van HW is mogelijk



Meer over MeArm

- Open hardware robot arm
- Tekeningen zijn beschikbaar op internet en kan gemakkelijk in het Stadslab met de lasersnijder uitgesneden worden (hout, plexiglas, 3mm)
- Kan bediend worden via Arduino (schemas ook op internet beschikbaar)



Ontwikkelomgeving

- Arduino met behulp van de library Cantino
 - <https://github.com/QQuick/Cantino>
- Online (zonder installatie) ontwikkelomgeving:
 - <https://repl.it/languages/cpp>
- Aanbevolen text editor :
 - Notepad++ (syntax highlight)
- Facultatief: MS Visual Studio Code
 - <https://code.visualstudio.com>

It's testing time!

Wie heeft er al ervaring met C++?

Ga naar: www.socrative.com ->login Student-> Room Name:

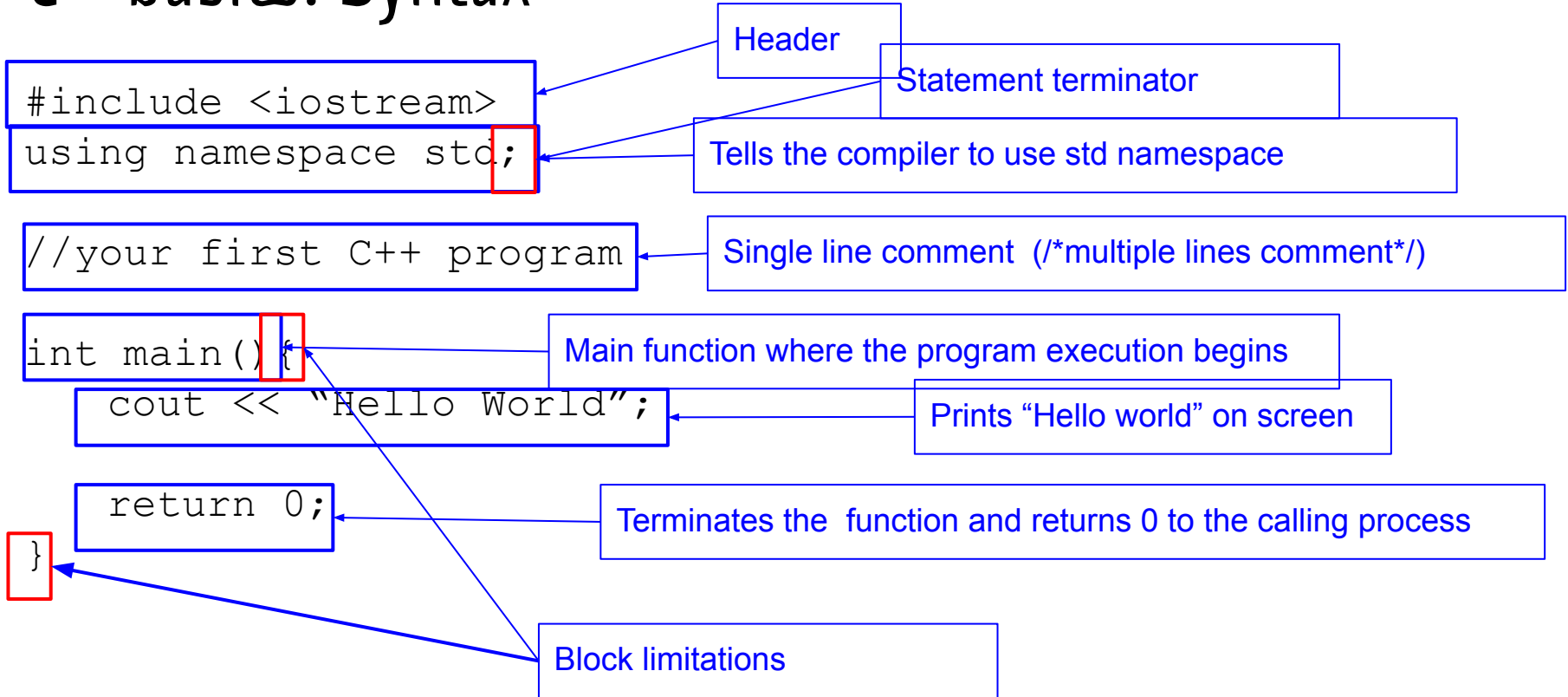
PIVAO123

Geschiedenis van C++

- Uitgevonden door Bjarne Stroustrup
- Oktober 1985: eerste commerciële implementatie werd uitgebracht
- 1989: versie 2.0 van C++
- 1991: geactualiseerde tweede editie van The C++ Programming Language
- 1998: C++ 98, derde editie van The C++ Programming Language
- 1999-2010: kleine revisies
- 2011: C++ 11, grote aanpassingen (taal en library toevoegingen)
- 2014: C++ 14, taal en library aanpassingen
- 2017: C++ 17, grote aanpassingen, ISO standard

Bron: <http://en.cppreference.com/w/cpp/language/history>

C++ basics: Syntax



Try it out!

- Schrijf je eigen eerste C++ programma die het volgende op het scherm toont:

<Naam> <Achternaam> <studentnummer>

<Woonplaats>

-->Let op! Er is een "new line"

- Zelf uitzoeken: syntax voor "new line" en "end line"
- Toepassen van "new line" en "end line" in je programma

Try it out!

- Herschrijf hetzelfde programma als hiervoor maar dan in Arduino omgeving
- Maak hierbij gebruik van de cantino library

Data types

Type	Keyword
Boolean	bool
Character	char
Integer	int
Floating point	float
Double floating point	double
valueless	void

Signed, unsigned, short, long

Try it out!

- Alle data types hebben een grootte. Print op het scherm de grootte van minimaal 6 datatypes door gebruik te maken van het commando **sizeof(<datatype>)**. Experimenteer met verschillende datatypes door de basistypes te combineren met Signed, unsigned, short, long
- Gebruik hiervoor je vorige programma of maak een nieuwe

Variabelen

```
#include <iostream>
using namespace std;

// Global variable definitions and (optional) initialisations
int
    startJaar = 1940,
    aantalJaren = 5;
int eindJaar = 0;

int main () {
    float gemiddeldeCijfer;           // Local variable definition
    eindJaar = startJaar + aantalJaren; // assignment
    cout << eindJaar << endl ;

    gemiddeldeCijfer = 70.0/3;        // assignment
    cout << gemiddeldeCijfer << endl ;
    return 0;
}
```

Try it out!

- Schrijf een programma die het volgende doet:
 - vraagt hoe oud je bent,
 - Wacht op input van een nummer
 - Het ingevoerde nummer opslaat in een variabele met een betekenisvolle naam
 - Het ingevoerde nummer weer op scherm print
- Tips:
 - `cout<<`
 - `cin>>`

Try it out!

- alle data types hebben een grootte. Print op het scherm de grootte van minimaal 6 datatypes door gebruik te maken van het commando **sizeof(<datatype>)**. Experimenteer met verschillende datatypes door de basistypes te combineren met Signed, unsigned, short, long
- Gebruik hiervoor je vorige programma of maak een nieuwe

arrays

```
float temperature[3];           // array of floats:
                                // temperature[0] etc.

char productNumber[5];         // an array of 5 chars
char choice[2]={'a','b'};      // Initialized array of char

int nrOfEyesOfDice[6]={1,2,3,4,5,6}; // Initialized array of int
int inputFieldMatrix[4][4];      // Twodimensional array
```

Try it out!

- Schrijf een programma dat het volgende doet:
 - Een array van integers declareert
 - De gebruiker vraagt hoeveel elementen in het array moeten komen (max 100)
 - De array vult met random getallen tussen 1 en 100
 - Het hoogste getal berekent en uitprint
- Tips:
 - Een array doorloop je d.m.v. een index
 - De positie van het eerste element in een array heeft index 0
 - Random integers tussen 1 en 100 genereer je met `rand() % 100+1`

Voorbeeld van uitwerking

Voorbeeld van uitwerking

[Repl.it arrayrandom](#)

Operators

Let's play Kahoot

arithmetic operators

Operator	Description	Example (given int a = 10, b = 20)
+	addition	<code>a + b == 30</code>
-	Subtraction	<code>a - b == -10</code>
*	Multiply	<code>a * b == 200</code>
/	Divide	<code>a / b == 0 (!)</code>
%	Modulo	<code>a % b == 10</code>
++	Increment	<code>teller++</code>
--	Decrement	<code>countdown--</code>

Relational operators: true or false?

Operator	Description	Example (given a=10,b=20)
==	Check if operands are equal	(a == b) == false
!=	Check if operands are not equal	(b != a) == true
>	Check if left operand is greater than right operand	(a > b) == false
<	Check if left operand is smaller than right operand	(a < b) == true
>=	Check if left operand is greater or equal than right operand	(a >= b) == false
<=	Check if left operand is smaller or equal than right operand	(a <= b) == true

Logical operators: true or false?

Operator	Description	Example (given a=false, b=true)
&&	Logical AND	(a && b) == false
	Logical OR	(a b) == true
!	Logical NOT	!(a && b) == true

true == 1 (eigenlijk alles wat niet 0 is wordt behandeld als true)

false == 0

bitwise operators: huiswerk 1

Gegeven $a=60$ en $b=13$, zet deze om in binair en bereken de resultaten van onderstaande operatoren. Schrijf het resultaat binair maar ook als int

Operator	Description	Example (given int $a=60, b=3$)
&		$a \& b ==$
		$a b ==$
^		$a \wedge b ==$
~		$\sim b ==$
<<		$a << b ==$
>>		$a >> b ==$

Assignment operators

```
1 #include <iostream>
2 using namespace std;
3
4 main() {
5
6     int a = 21;
7     int c = a;
8
9     c += a;
10    c -= a;
11    c *= a;
12    c /= a;
13
14    c =200;
15
```

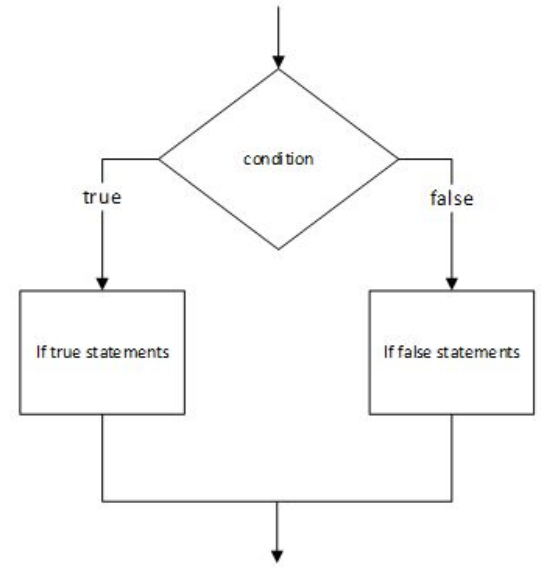
```
16    c%= a;
17    c<<= 2;
18    c>>= 2;
19    c&= 2;
20    c^= 2;
21    c|= 2;
22    return 0;
23 }
24
25
```

Statements: if

if (condition) statement else statement

```
if (age < 40 ) {  
    cout << "lekker jong nog" << endl;  
}
```

```
if (age < 18) {  
    cout << "Sorry, " << age << " years is too young to smoke " << endl;  
}  
else {  
    cout << "Sorry, " << age << " years is too old for a pacifier!" << endl;  
}
```



Try it out!

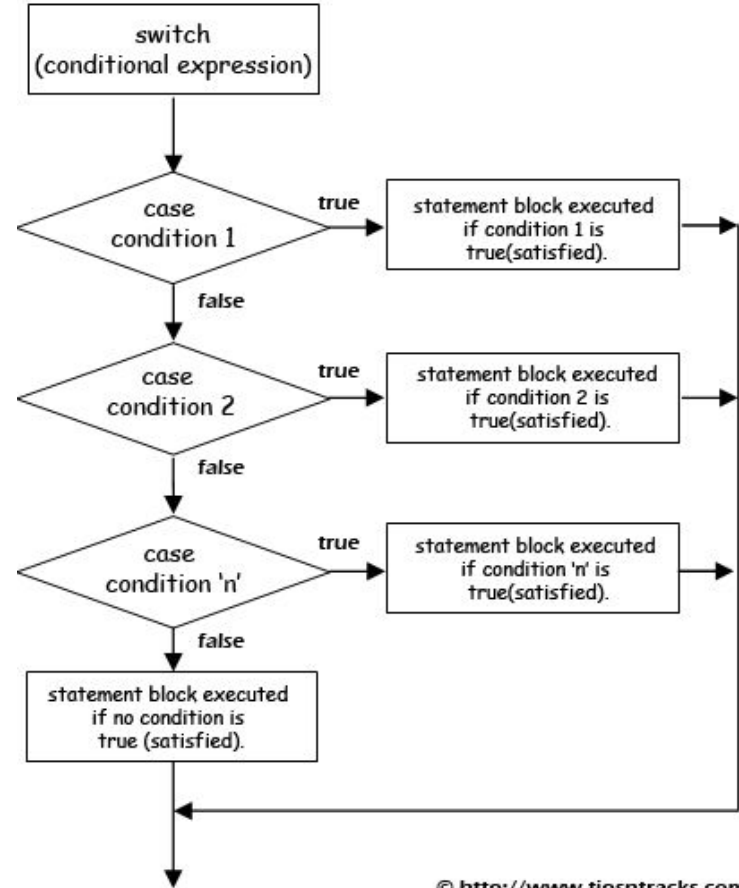
- Pas het vorige programma dat naar de leeftijd vraagt aan en zorg dat het het volgende doet:
 - “Je bent een kind” op het scherm uitprint als de leeftijd tussen 0 en 10 is
 - “Je bent een puber” op het scherm uitprint als de leeftijd tussen 10 en 18 is
 - “Je bent een volwassene” op het scherm uitprint als de leeftijd meer dan 18 is
- Tips:
 - Gebruik `&&` (AND), `||` (OR) indien je dat wil
 - Geneste if statements zijn mogelijk

Statements: switch

//discretevariable= int of char of enum , ...

```
switch (discretevariable) {  
    case discretevariable: {statements;}  
    ...  
    default: {statements;}  
}
```

```
switch (age){  
    case 0 : { cout<<"you're a baby";break;}  
    case 18: { cout<<"you're a grown  
up";break;}  
    case 80: { cout<<"you're old!";break;}  
    default: { cout<<"you're "<< age << " years  
old";}  
}
```



Try it out!

- Schrijf een programma om een simpele rekenmachine te realiseren:
 - Vraagt de gebruiker om een operand in te voeren (+, -, *, /)
 - Vraagt de gebruiker om twee cijfers in te voeren
 - De gekozen operatie tussen de twee cijfers uit te voeren en op scherm weer te geven
- Tips:
 - Gebruik switch
 - Gebruik break

Voorbeeld van uitwerking

Voorbeeld van uitwerking

[Repl.it switch.cpp](#)

Statements: for

for(initializer; testExpression; updater) statement

```
int number;
```

```
cout << "Enter a positive integer: ";  
cin >> number;
```

```
for (int index = 0; index < number; index++) {  
    cout << index << endl;  
}
```

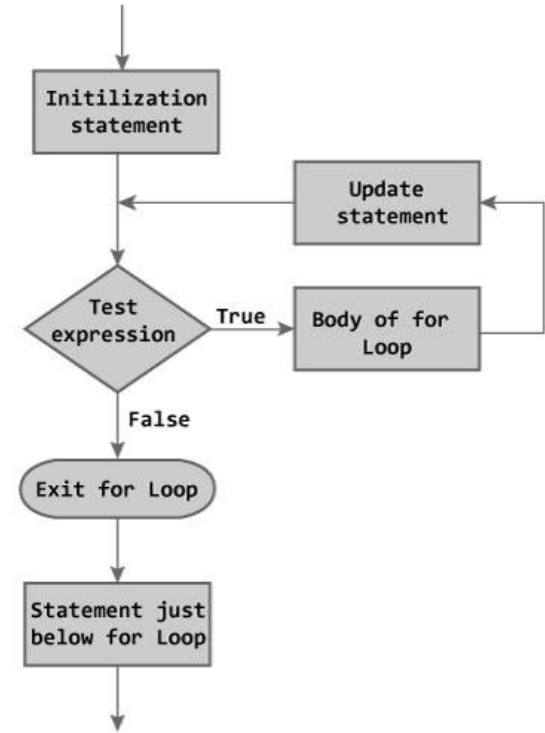


Figure: Flowchart of for Loop

Statements: while

while (condition) statement

```
int number = 0, factor = 1, factorial = 1;
```

```
cout << "Enter a positive integer: ";  
cin >> number;
```

```
while ( factor <= number) {  
    factorial *= factor; // Of factor++  
    factor++;  
}
```

```
cout<<"Factorial of "<< number <<" = "<< factorial;
```

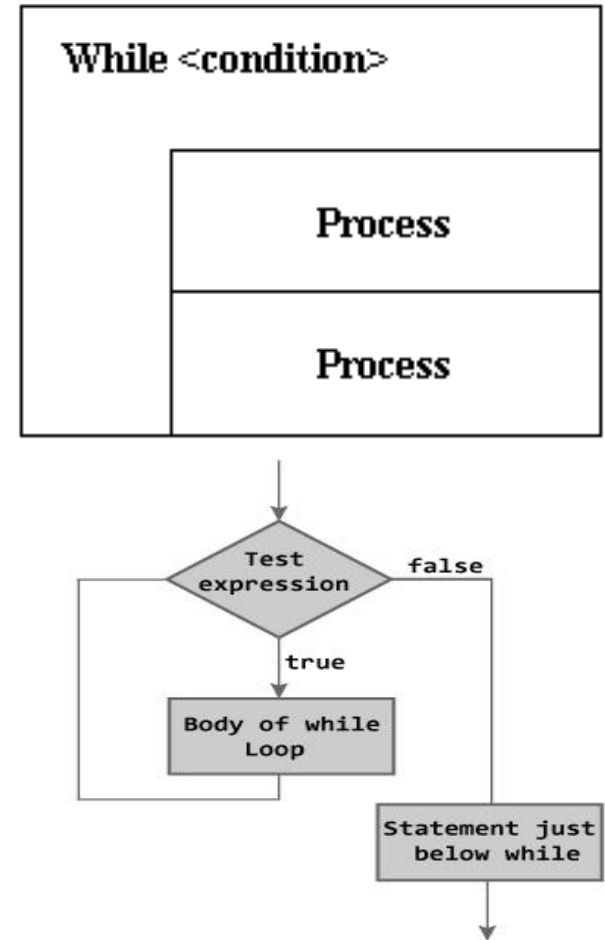


Figure: Flowchart of while Loop

Try it out!

- Maak een programma dat het volgende doet:
 - Vraagt de gebruiker om een getal (int number) in te voeren
 - Op het scherm de waarde van $1+2+3+\dots+n$ weergeeft
- Eis: het programma moet 2x hetzelfde uitvoeren met een for loop en met een while loop

- Tips:
 - Let op de volgorde van operators zoals ++

Voorbeeld van uitwerking

Voorbeeld van uitwerking

[Repl.it forLoop.cpp](#)

Try it out!

- Schrijf een programma dat het volgende doet:
 - Laat de gebruiker een zin typen die met een "." moet eindigen
 - Telt hoeveel keer "a" in de zin voorkomt
- Tips:
 - Gebruik char
 - Gebruikt een counter

Voorbeeld van uitwerking

Voorbeeld van uitwerking

[Repl.it whileLoop.cpp](#)

Statements: do while

do statement while(condition)

```
int number, sum = 0;
```

```
do {  
    cout<<"Enter a number: ";  
    cin>>number;  
    sum += number;  
}  
while(number != 0);  
cout<<"Total sum = " <<sum << endl;
```

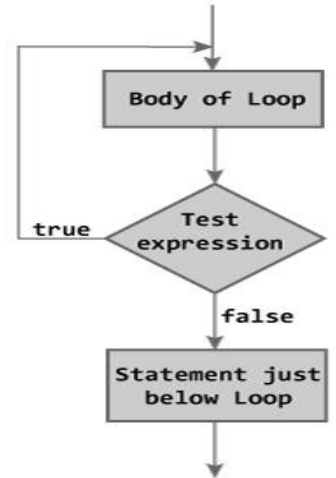


Figure: Flowchart of do...while Loop

Try it out!

- Maak een programma dat het volgende doet:
 - Laat de gebruiker een (lang) getal invoeren (b.v.121)
 - De volgorde van de cijfers omkeert
 - Checkt of dit een palindroom is (palindroom is een nummer die ook omgekeerd hetzelfde is, zoals 121)
- Tips:
 - Gebruik do while
 - Operator % (modulo) kan handig zijn

Voorbeeld van uitwerking

Voorbeeld van uitwerking

[Repl.it dowhileLoop.cpp](#)

Try it out!

- Use your creativity to create and initialize your struct
- Tips:
 - be creative!

Enumeration

- Het is een user-defined datatype dat bestaat uit een reeks van constanten

```
enum Season { spring, summer, autumn, winter };
```

- by default, de eerste waarde van de reeks krijgt 0 als waarde, de tweede 1, en zo voort. Dit kan veranderd worden:

```
enum Season  
{  
    spring = 0,  
    summer = 4,  
    autumn = 8,  
    winter = 12  
};
```

Enumerations: voorbeeld

```
#include <iostream>
using namespace std;

enum Week { sunday, monday, tuesday, wednesday, thursday, friday, saturday
};

int main()
{
    Week today; // of Week today= wednesday;
    today = wednesday;
    cout << "Day " << today+1;
    return 0;
}
```

Try it out!

- Zie huiswerk

Structure

Struct is een verzameling van variabelen met verschillende types onder een naam

```
struct Person
{
    char name[50];
    int age;
    float salary;
};
```


Struct: voorbeeld

```
#include <iostream>
using namespace std;
```

```
struct Person{
    char name[50];
    int age;
    float salary;
};
```

```
int main(){
    Person person1;

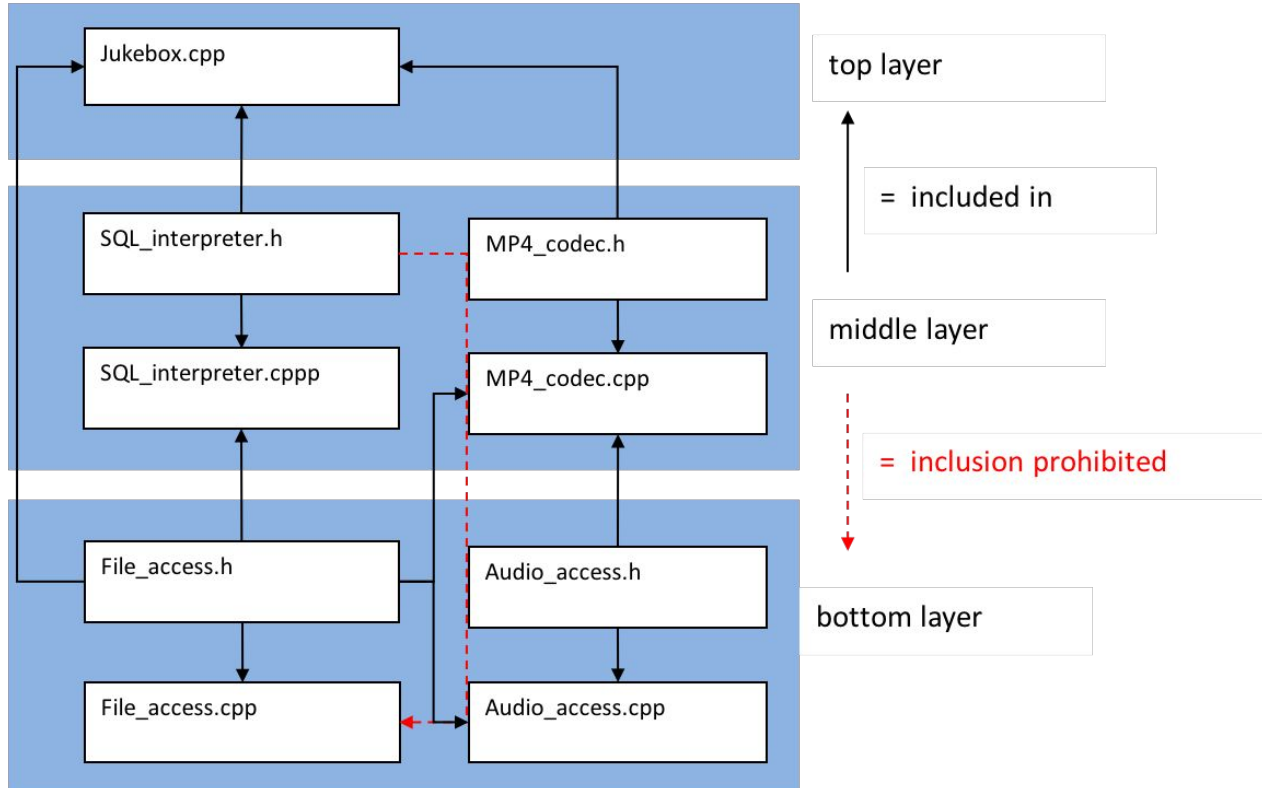
    person1.age = 30;
    person1.salary = 2000;

    cout << "What is your name?"<<endl;
    cin >> person1.name;

    cout <<"Hello " << person1.name<< " age: " << person1.age << endl;
    cout << "Salary: " << person1.salary << endl;

    return 0;
}
```

File structuur



Headers: waarom?

```
// Definition in library module
```

```
void greet (int i) {
```

```
    cout << "Hello, Rotterdam " << i << endl ();
```

```
}
```

```
// Use in main
```

```
if (a > b) {
```

```
    greet (3.5)
```

Headers

```
#include <iostream>    // Standard include path  
#include "module.h"    // Current directory
```

- In this course only class declarations allowed in header
- ☐ No global #defines, constants, variables c.q instances
- ☐ No free functions



Huiswerk

- „ Oefen door programma's te maken die het volgende doen:
 - De tafel berekenen en uitprinten van een getal door de gebruiker gegeven (b.v. input:5 zal weergeven: $5 \times 1 = 5$, $5 \times 2 = 10$, ..., $5 \times 10 = 50$)
 - berekent of een door de gebruiker ingegeven letter een klinker of medeklinker is
 - De fibonacci serie print tot een door de gebruiker gegeven cijfer (fibonacci 0,1,1,2,3,5...elke volgende cijfer is de som van de voorgaande twee cijfers)
 - arduino opdracht: koppel drie ledjes aan je arduino (rood, geel groen), laat de gebruiker een kleur kiezen en zet de bijbehorende led aan (gebruik enum)