C++ifPCwhileLinuxdiffintFirefoxboolpublicprivateWindows

fileg++UNIXVisualOSFedoralscharpsopenclosegetputarray

Maccpelsecoutmain RedHatmkdireditorstructDebianSublime

classusingcmathChrometruecpSUSEnamespaceenuminclude

doublecdGNUdokillobjectcompilermorelinkiostreamcinnotlpr

WWWmakefilegrepiomanipgeditfloatGNOMEfstreamrmfalse

andorCode::Blockstopforstringe-mailmanchmodWSLUbuntu

Programmeermethoden

Walter Kosters en Jonathan Vis

week 1: 4–8 september 2023

Introductie

www.liacs.leidenuniv.nl/~kosterswa/pm/

↑

tilde (~)

E-mailadres: pm@liacs.leidenuniv.nl

Het college wordt gegeven door dr. W.A. (Walter) Kosters en dr. J.K. (Jonathan) Vis.

Werkcolleges worden verzorgd door studentassistenten:
Donderdag: Perri van den Berg, Teun Bergsma, Amber van den Broek, Julian Poelsma en Justin de Rooij.
Vrijdag: Zahir Bingen, Sem Kluiver, Xander Lenstra, Leon Monster en Daan van Vliet.

Anderen geven PM-Python voor Bioinformatica, DSAI en Informatica & Economie; en voor Natuur/Sterrenkunde.

Er zijn verschillende soorten activiteiten:

colleges live in Gorlaeus 1; zie ook (oudere) video's:
www.liacs.leidenuniv.nl/~kosterswa/pm/videos.php

werkcolleges in (computer)zalen Snellius; handig: laptop mee; donderdagmiddag of vrijdagochtend; er is een aanwezigheidscontrole!

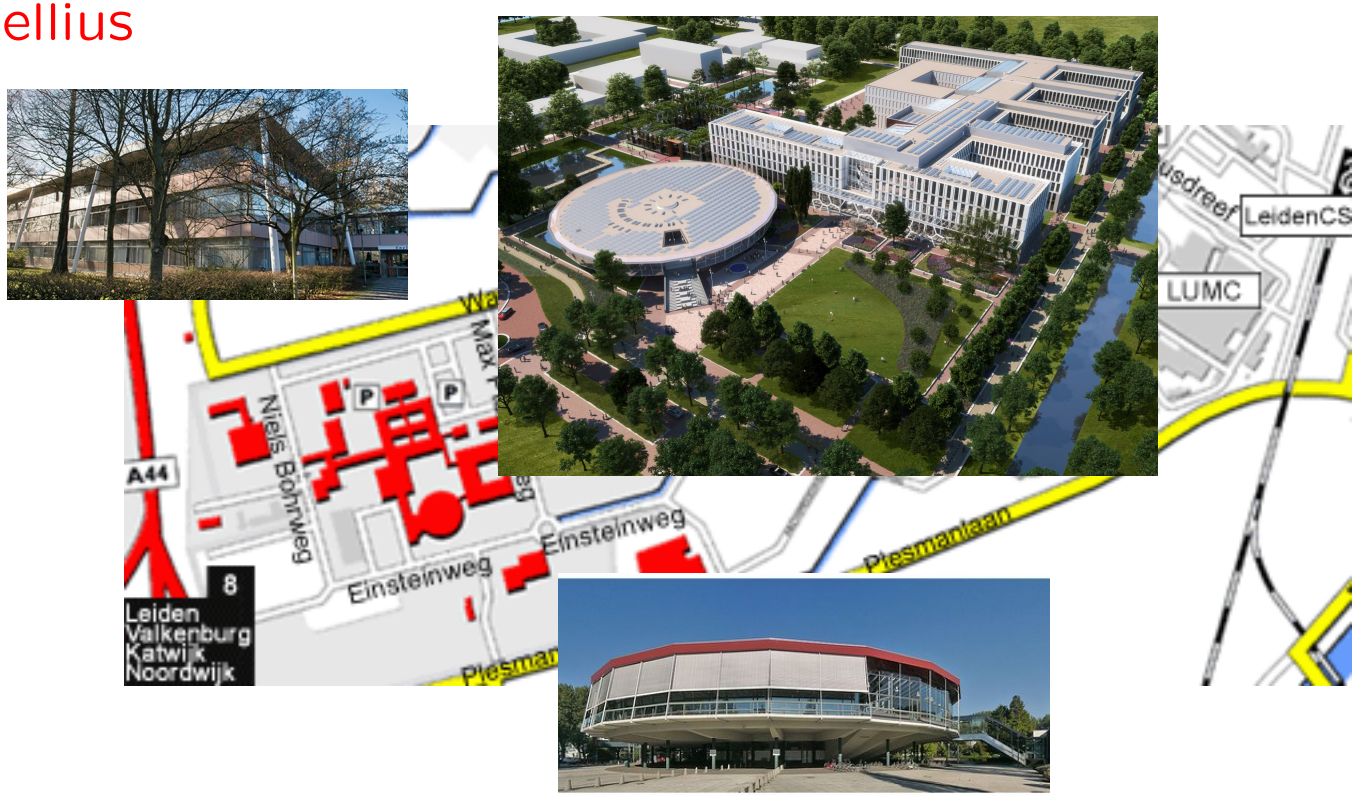
vragenuren facultatief; in Snellius, direct na werkcollege

Eerste week: 4–8 september 2023; laatste week: 11–15 december. Geen activiteiten in de week van 23–27 oktober.

www.liacs.leidenuniv.nl/~kosterswa/pm/schema.php

ma	di	wo	donderdag	vrijdag
			11:00–12:45 Gorlaeus zaal 1 college 1 iedereen	11:00–12:45 Snellius 303/7/9 werkcollege 1, vragenuurtje Wiskundigen
			13:15–. . . Snellius 302/3/6/7/9 werkcollege 1, vragenuur Informatici	

Snellius



Gorlaeus

Iedereen gebruikt zijn/haar ULCN-account. Daarmee kun je ook WiFi gebruiken in universiteitsgebouwen, via `eduroam` of `leidenuniv`. Zet een "forward" voor e-mails!

Je krijgt daarmee ook een *home directory* ("homedir") in het universitaire UNIX-systeem. In de computerzalen van het Snellius kun je daar via Linux mee werken.
Hoe kom je vanaf huis bij je UNIX-files? Antwoord: "ssh-en" en "scp-en" naar

`a.einstein@sshgw.leidenuniv.nl`

(en verder) als je `a.einstein` bent. Zie later.

Sluw: laptop, USB, eigen e-mail, de cloud, . . .

Er moeten vier programmeeropgaven gemaakt worden. **Als** ze alle voldoende zijn (hooguit één 5) **en** het tentamen voldoende ($\geqslant 5.5$) is gemaakt, krijg je zes studiepunten:

$$\text{Eindcijfer} = \frac{2 \times \text{Schriftelijk} + \frac{\text{Op}_1 + \text{Op}_2 + \text{Op}_3 + \text{Op}_4}{4}}{3}$$

(afgerond naar het dichtstbijzijnde element uit de verzameling $\{1, 2, 3, 4, 5,\ 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10\}$).

Heb je nog deelresultaten uit voorgaande jaren? Ga langs bij de docent!

Cijfers: Brightspace (meld je aan = "enroll").

- in tweetallen maken
- wel overleggen en om hulp vragen, niet kopiëren;
  zie website voor huisregels en `richtlijnen`, bijvoorbeeld
  voor aanvullen, . . .
- `op tijd` inleveren: deadlines (-1 per week te laat)
- deels maken tijdens werkcolleges, deels thuis
- vragenuren!

- Opgave 1: maandag 25 september 2023, 18:00 uur
- Opgave 2: maandag 16 oktober 2023, 18:00 uur
- Opgave 3: maandag 13 november 2023, 18:00 uur
- Opgave 4: maandag 11 december 2023, 18:00 uur

Lever per team (“group”) de C++-code in Brightspace in.
En het geprinte verslag in de doos bij Snellius, kamer 159.

Programma 2023 — Eerste helft

week	onderwerp	boek	"dictaat"
4–8 sep	Introductie	1	1,2
11–15 sep	Types		3.1/3,3.9, op1/5
18–22 sep	Controlestructuren	2	3.4/5,op6/10
25–29 okt	Functies, files	3,4, 12.1/2	3.6/7,op11/17
2–6 okt	idem, vervolg		
9–13 okt	idem, Life	6,7.1	3.11,op18/25
16–20 okt	OOP, arrays	5	3.8,op26/30
...			

op = "papieren" opgaven van website ("Handouts"); zelf maken, antwoorden: zie website.

In rood: de weken met een deadline op de maandag erna.

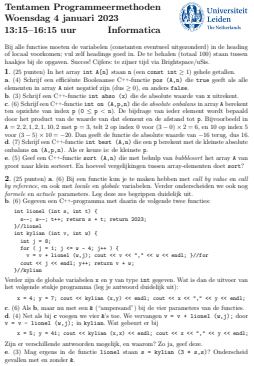
Het schriftelijk tentamen is op

- woensdag 17 januari 2024, 13:00–16:00 uur
  Universitair Sportcentrum
- hertentamen: donderdag 28 maart 2024, 13:00–16:00 uur
  Snellius

Zie de website voor oude tentamens en uitwerkingen.

Tijdig aanmelden via uSis! En Brightspace (tegelijk) . . .

[4–1–2023](#) ⟶



Tentamen Programmeermethoden
Woensdag 4 januari 2023
13:15–16:15 uur **Informatica**

Universiteit Leiden
The Netherlands

Bij alle functies moeten de variabelen (constanten eventueel uitgezonderd) in de kopregel of lokaal voorkomen; vul zelf benodigde goed in. De te behalen (totaal 100) staan tussen haakjes bij de opgaven. Succes! Cijfers: te zijner tijd via Brightspace/uSis.

1. (25 punten) In het array `int A[n]` staan n (een `const int` ≥ 1) gehele getallen.
a. (4) Schrijf een efficiënte Booleanse C++-functie `pos (A,n)` die `true` geeft als alle elementen in array `A` niet negatief zijn (dus ≥ 0), en anders `false`.
b. (3) Schrijf een C++-functie `int abes (x)` die de absolute waarde van `x` uitrekent.
c. (6) Schrijf een C++-functie `int on (A,p,n)` die de *absolute eenheime* in array `A` berekent ten opzichte van index `p` ($0 \leq p < n$). De bijdrage van ieder element wordt bepaald door het product van de waarde van dat element en de afstand tot `p`. Bijvoorbeeld in `A` = 2, 2, 1, 2, 1, 10, 2 met `p` = 3, heb 2 op index 0 voor $(3-0) \times 2 = 6$, en 10 op index 5 voor $(3-5) \times 10 = -20$. Dan geeft de functie de absolute waarde van -16 terug, dus 16.
d. (7) Schrijf een C++-functie `int best (A,n)` die een `p` berekent met de kleinste absolute eenheime `on (A,p,n)`. Als er leuze ie die kleinste `p`.
e. (5) Geef een C++-functie `sort (A,n)` die met behulp van *bubblesort* het array `A` van groot naar klein sorteert. En hoeveel vergelijkingen tussen array-elementen doet `sort`?

2. (25 punten) **a**. (6) Bij een functie kun je te maken hebben met *call by value* en *call by reference*, en ook met *locale* en globale variabelen. Verder onderscheiden we ook nog *formele* en *actuele* parameters. Leg deze zes begrippen duidelijk uit.
b. (6) Gegeven een C++-programma met daarin de volgende twee functies:

```
int lionel (int x, int t) {
    x--; x--; t++; return x + t; return 2023;
}//lionel
int kylian (int v, int w) {
  int j = 0;
  for ( ; j < w - 4; j++ ) {
    v = v + lionel (w,j); cout << v << "," << w << endl; }//for
  cout << j << endl; j++; return v + w;
}//kylian
```

Verder zijn de globale variabelen x en y van type `int` gegeven. Wat is dan de uitvoer van het volgende stukje programma (leg je antwoord duidelijk uit):

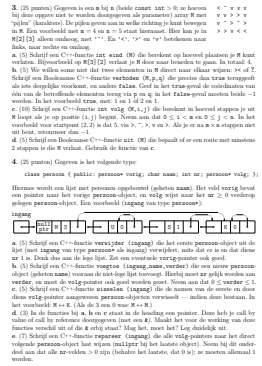
```
x = 4; y = 7; cout << kylian (x,y) << endl; cout << x << "," << y << endl;
```

c. (6) Als `b`, maar nu met een `&` ("ampersand") bij de vier parameters van de functies.
d. (4) Net als bij **c** vroegen we vier `&`'s toe. We vervangen `v = v + lionel (w,j);` door `v = v - lionel (w,j);` in `kylian`. Wat gebeurt er bij

```
x = 5; y = 41; cout << kylian (x,x) << endl; cout << x << "," << y << endl;
```

Zijn er verschillende antwoorden mogelijk, en waarom? Zo ja, geef deze.
e. (3) Mag regeine in de functie `lionel` staan `z = kylian (3 * a,z)`? Onderscheid gevallen met en zonder `&`.

1



3. (25 punten) Gegeven is een `n` bij `n` (beide `const int` > 0; ze hoeven bij deze opgave niet te worden doorgegeven als parameter) array `M` met "pijlen" (karakters). De pijlen geven aan in welke richting je kunt bewegen in `M`. Een voorbeeld met `n` = 4 en `n` = 5 staat hiernaast. Hier kun je in `M[2][2]` alleen omhoog, met `'^'`. En `'<'`, `'>'` en `'v'` betekenen naar links, naar rechts en omlaag.

```
< ^ v v v
v v > > v
v ^ > ^ >
> > v < <
```

a. (5) Schrijf een C++-functie `int sind (M)` die berekent op hoeveel plaatsen je `M` kunt verlaten. Bijvoorbeeld op `M[3][2]` verlaat je `M` door naar beneden te gaan. In totaal: 4.
b. (5) We willen soms niet dat twee elementen in `M` direct naar elkaar wijzen: `><` of `^v`. Schrijf een Booleanse C++-functie `verboden (M,p,q)` die precies dan `true` teruggeeft als iets dergelijks voorkomt, en anders `false`. Geef in het `true`-geval de coördinaten van één van de betreffende elementen terug via `p` en `q`; in het `false`-geval moeten beide -1 worden. In het voorbeeld `true`, met: 1 en 1 of 2 en 1.
c. (10) Schrijf een C++-functie `int volg (M,i,j)` die berekent in hoeveel stappen je uit `M` loopt als je op positie (i, j) begint. Neem aan dat $0 \leq i < n$ en $0 \leq j < n$. In het voorbeeld voor startpunt $(2, 2)$ is dat 5, via $>, ^, >$ en $>$. Als je ze aan `n` x stappen niet uit hoeft, retourneer dan -1.
d. (5) Schrijf een Booleanse C++-functie `uit (M)` die bepaalt of er een route met minstens 2 stappen is die `M` verlaat. Gebruik de functie van **c**.

4. (25 punten) Gegeven is het volgende type:

```
class persoon { public: persoon* volg; char naam; int nr; persoon* volg; };
```

Hiermee wordt een lijst met personen opgebouwd (geheten `naam`). Het veld `volg` bevat een pointer naar het vorige `persoon`-object, en volg wijst naar het nr ≥ 0 vorenliep gelegen `persoon`-object. Een voorbeeld (`ingang` van type `persoon*`):

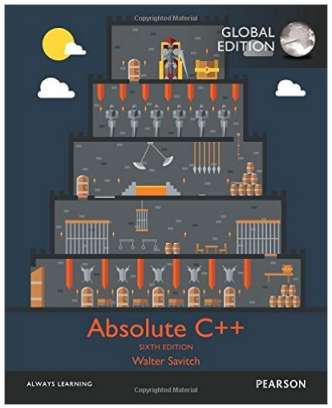
a. (5) Schrijf een C++-functie `vorei_jder (ingang)` die het eerste `persoon`-object uit de lijst (met `ingang` van type `persoon*` als ingang) vervrijdert, zulte dat er is en dat diens `nr` 1 is. Denk dus aan de lege lijst. Zet een eventuele `volg`-pointer ook goed.
b. (5) Schrijf een C++-functie `voegtoe (ingang,naam,verder)` die een nieuw `persoon`-object (geheten `naam`) vooraan de niet-lege lijst toevoegt. Hierbij moet `nr` gelijk worden aan `verder`, en moet de `volg`-pointer ook goed worden gezet. Neem aan dat $0 \leq$ `verder` ≤ 1.
c. (5) Schrijf een C++-functie `wisselen (ingang)` die de namen van de eerste en door diens `volg`-pointer aangewezen `persoon`-objecten verwisselt — indien deze bestaan. In het voorbeeld: `N` $\leftrightarrow$ `K`. (Als de 3 een 0 was: `N` $\leftrightarrow$ `R`.)
d. (3) In de functies bij **a**, **b** en **c** staat in de kopregel een pointer. Dient hoh je call by value of call by reference doorgegeven (met een `&`). Maakt het voor de werking van deze functies verschil uit of die `&` erbij staat? Mag het, moet het? Leg duidelijk uit.
e. (7) Schrijf een C++-functie `reparseer (ingang)` die alle `volg`-pointers naar het direct volgende `persoon`-object laat wijzen (`nullptr` bij het laatste object). Neem bij dit onderdeel aan dat alle `nr`-velden > 0 zijn (behalve het laatste, dat 0 is); ze moeten allemaal 1 worden.

2

We maken gebruik van het volgende boek:

W. Savitch
Absolute C++
sixth edition
Addison-Wesley, 2016

Oudere drukken zijn ook goed (ouderejaars!). En er zijn vele andere boeken.



En het "dictaat" (hier) en de sheets (daar).

Naast Microsoft Windows kunnen PCs ook draaien onder Linux, een operating systeem (OS) uit de UNIX-wereld (vergelijk Android). In Windows 10 en 11 heb je Windows Subsystem for Linux (WSL) — en dat gebruiken we.

(Met een Ubuntu Live-CD/DVD/USB
start je PC meteen op in GNOME,
een grafische windows-omgeving
bovenop Linux.
Beter: dual boot.
Let er op dat je op een verstandige
plaats (USB/e-mail) moet saven.

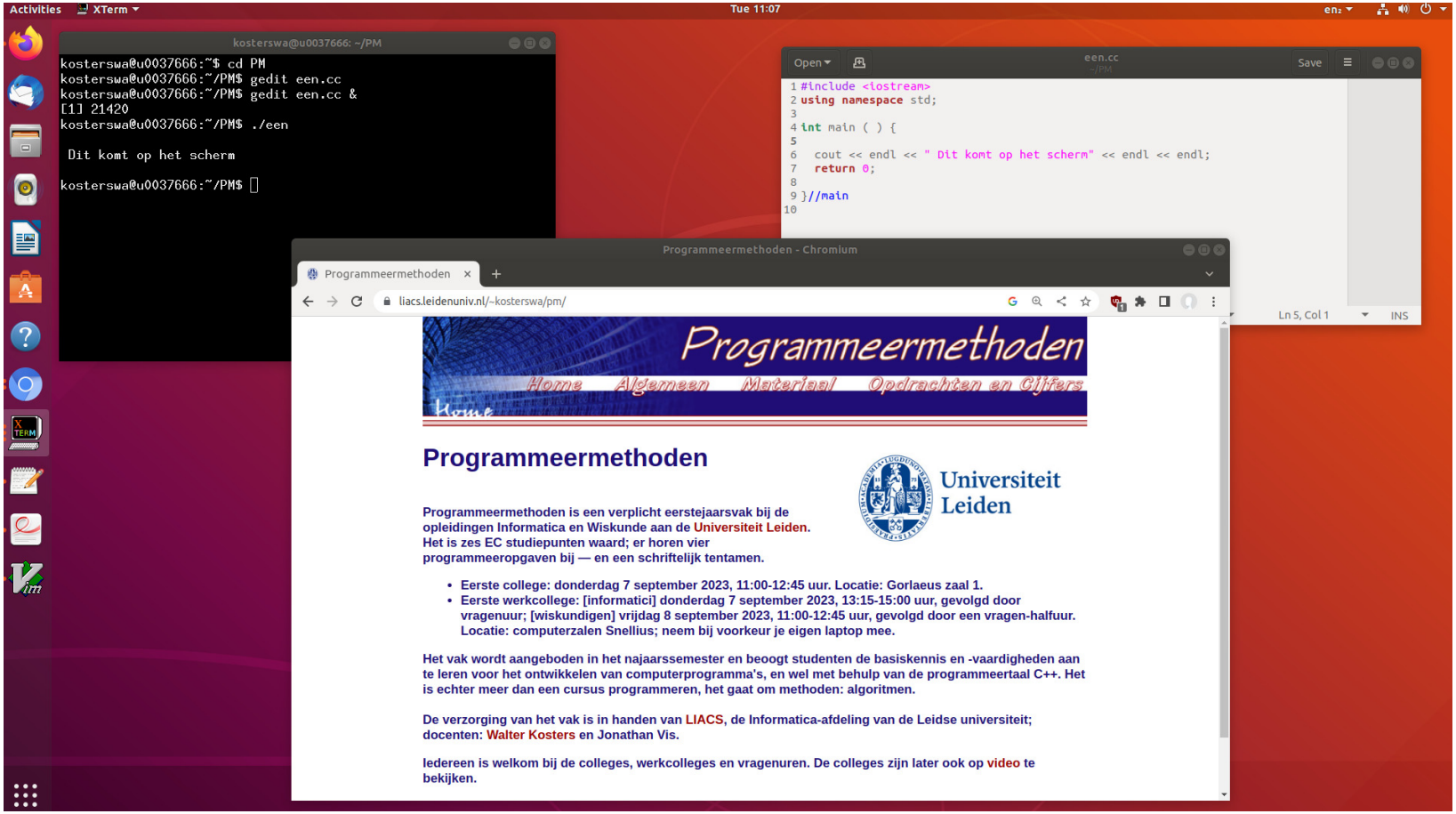
Je kunt ook "gratis" distributies als SUSE, Fedora, Debian, . . . gebruiken — maar dan moet je er meer vanaf weten.)

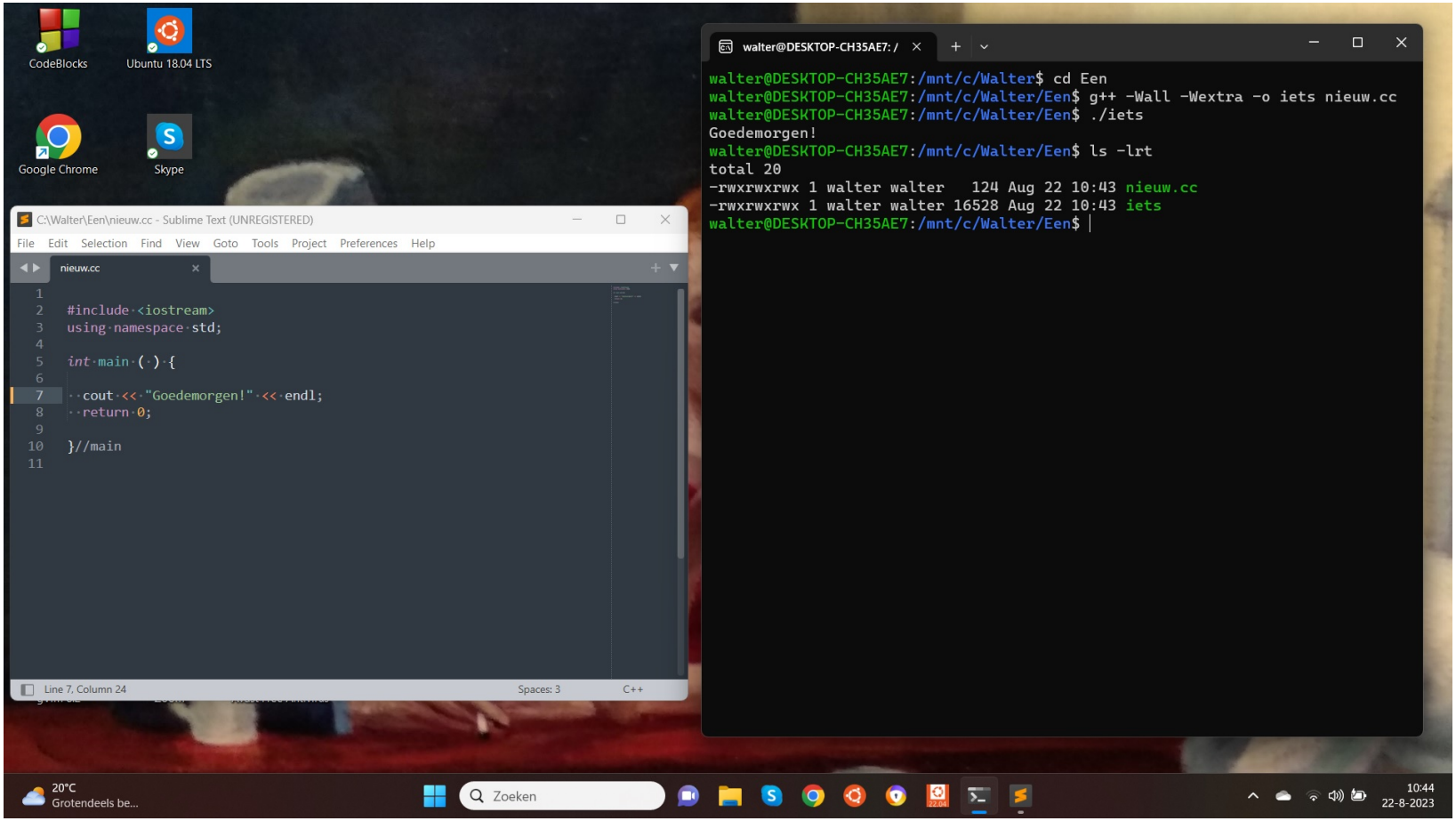
Als je thuis gratis C++ wilt doen, zijn de mogelijkheden:

- Windows: installeer Windows Subsystem for Linux (WSL, zie straks) en eventueel editor Sublime Text
- Mac: zie hier

  Mac-video
- Linux: haal Ubuntu van www.ubuntu.org
  huidige versie: 23.04 (in computerzalen: 18.04)
  gebruik editor `gedit` en compiler `g++`

  Linux-video
- Windows: Code::Blocks, zie straks


Activities
XTerm
Tue 11:07
kosterswa@u0037666: ~/PM
kosterswa@u0037666:~$ cd PM
kosterswa@u0037666:~/PM$ gedit een.cc
kosterswa@u0037666:~/PM$ gedit een.cc &
[1] 21420
kosterswa@u0037666:~/PM$ ./een
Dit komt op het scherm
kosterswa@u0037666:~/PM$
Open
een.cc
Save
#include <iostream>
using namespace std;
int main ( ) {
cout << endl << " Dit komt op het scherm" << endl << endl;
return 0;
}//main
Ln 5, Col 1
INS
Programmeermethoden - Chromium
Programmeermethoden
liacs.leidenuniv.nl/~kosterswa/pm/
Programmeermethoden
Home Algemeen Materiaal Opdrachten en Cijfers
Programmeermethoden
Universiteit Leiden
Programmeermethoden is een verplicht eerstejaarsvak bij de opleidingen Informatica en Wiskunde aan de Universiteit Leiden. Het is zes EC studiepunten waard; er horen vier programmeeropgaven bij — en een schriftelijk tentamen.
Eerste college: donderdag 7 september 2023, 11:00-12:45 uur. Locatie: Gorlaeus zaal 1.
Eerste werkcollege: [informatici] donderdag 7 september 2023, 13:15-15:00 uur, gevolgd door vragenuur; [wiskundigen] vrijdag 8 september 2023, 11:00-12:45 uur, gevolgd door een vragen-halfuur. Locatie: computerzalen Snellius; neem bij voorkeur je eigen laptop mee.
Het vak wordt aangeboden in het najaarssemester en beoogt studenten de basiskennis en -vaardigheden aan te leren voor het ontwikkelen van computerprogramma's, en wel met behulp van de programmeertaal C++. Het is echter meer dan een cursus programmeren, het gaat om methoden: algoritmen.
De verzorging van het vak is in handen van LIACS, de Informatica-afdeling van de Leidse universiteit; docenten: Walter Kosters en Jonathan Vis.
Iedereen is welkom bij de colleges, werkcolleges en vragenuren. De colleges zijn later ook op video te bekijken.


CodeBlocks
Ubuntu 18.04 LTS
Google Chrome
Skype
C:\Walter\Een\nieuw.cc - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
nieuw.cc
#include <iostream>
using namespace std;
int main(){
cout<<"Goedemorgen!"<<endl;
return 0;
}//main
Line 7, Column 24
Spaces: 3
C++
walter@DESKTOP-CH35AE7: /
walter@DESKTOP-CH35AE7:/mnt/c/Walter$ cd Een
walter@DESKTOP-CH35AE7:/mnt/c/Walter/Een$ g++ -Wall -Wextra -o iets nieuw.cc
walter@DESKTOP-CH35AE7:/mnt/c/Walter/Een$ ./iets
Goedemorgen!
walter@DESKTOP-CH35AE7:/mnt/c/Walter/Een$ ls -lrt
total 20
-rwxrwxrwx 1 walter walter 124 Aug 22 10:43 nieuw.cc
-rwxrwxrwx 1 walter walter 16528 Aug 22 10:43 iets
walter@DESKTOP-CH35AE7:/mnt/c/Walter/Een$
20°C
Grotendeels be...
Zoeken
10:44
22-8-2023

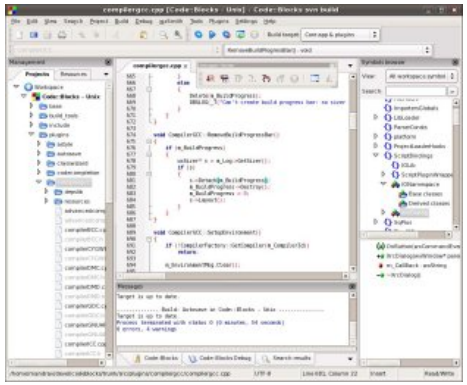
- Installeer Windows Subsystem for Linux (WSL) in Windows 11 (of 10), via `cmd`, `wsl --install -d Ubuntu-22.04`
- Open een terminal en installeer compiler `g++` met `sudo apt install g++` (doe altijd eerst `sudo apt update`).
- Gebruik een Windows-editor zoals `Sublime Text` of installeer in Linux `gedit`.
- Waar zijn je mappen = directories? Geef het commando `cd /mnt/c/naamvanmap` en zoeken maar . . .
- Hello world . . . , zie straks.

Je kunt ook Code::Blocks in Windows gebruiken, maar liever `Sublime Text` (of `gedit`) en `g++` in Linux (WSL).

video

- installeer Code::Blocks van `www.codeblocks.org`
  file: `codeblocks-20.03mingw-setup.exe`

- zet "warnings" aan!



- edit, compileer en run
  `helloworld.cc` (let op `.cc`)

- behandel een eventuele lastige firewall

Een werkend C++-programma maken gaat als volgt:

1. Tik in een editor C++-code, het "bron-programma".
2. Compileer (en link) dit met een compiler. Deze vertaalt C++-code naar machinetaal.
   Eventuele fouten: "compile-time-fouten" / "syntax-errors": "`inklude`" in plaats van "`include`".
3. Draai (= run) deze executable vanuit het OS.
   Eventuele fouten: "run-time-fouten": deel door 0.

Herhaal deze cyclus zo vaak als nodig.
In Code::Blocks vinden alle stappen plaats in een Graphical User Interface (GUI).

(Er zijn ook nog logische fouten.)

Een eerste C++-programma, `helloworld.cc` geheten:

```
#include <iostream>
using namespace std;
int main ( ) {
  cout << "Dag allemaal!" << endl;
  return 0;
}//main
```

Dit programma zet alleen een tekstje op het beeldscherm.

Let op de — vooral voor mensen nuttige — layout.
En op hoofdletters en kleine letters.

Een tweede C++-programma (met syntax-highlighting):

```
1  // dit is een simpel programma
2 #include <iostream>
3 using namespace std;
4
5 int main ( ) {
6   int getal = 42;   // een variabele declareren
7                     // en initialiseren
8   cout << "Geef een geheel getal .. ";
9   cin >> getal;
10  cout << "Kwadraat is: "
11       << getal * getal << endl;
12  return 0;
13 }//main
```

Nog een C++-programma

```
// Dit is een regel met commentaar ...
#include <iostream> // moet er altijd bij
using namespace std;
const double PIE = 3.14159; // een constante (of cmath)
int main ( ) {
  int straal;  // straal van de cirkel, geheel getal
  cout << "Geef straal, daarna Enter .. ";
  cin >> straal;
  if ( straal > 0 )
    cout << "Oppervlakte "
         << PIE * straal * straal << endl;

  else
    cout << "Niet zo negatief ..." << endl;
  cout << "Einde van dit programma." << endl;
  return 0;
}//main
```

Nog een C++-programma

```
// Dit is een regel met commentaar ...
#include <iostream> // moet er altijd bij
using namespace std;
const double PIE = 3.14159; // een constante (of cmath)
int main ( ) {
  int straal;  // straal van de cirkel, geheel getal
  cout << "Geef straal, daarna Enter .. ";
  cin >> straal;
  if ( straal > 0 ) {        // accolades nodig!
    cout << "Oppervlakte ";
    cout << PIE * straal * straal << endl;
  }//if
  else // hier accolades niet nodig, maar liever wel!
    cout << "Niet zo negatief ..." << endl;
  cout << "Einde van dit programma." << endl;
  return 0;
}//main
```

Eerste programmeeropgave

Programmeermethoden

Home Algemeen Materiaal Opdrachten en Cijfers

Opdrachten en Cijfers

Programmeermethoden 2023
Eerste programmeeropgave: Vermenigvuldiging

De *eerste* programmeeropgave van het vak **Programmeermethoden** in het najaar van 2023 heet *Vermenigvuldiging*; zie ook het **eerste**, **tweede** en **derde** werkcollege.

Deze opgave probeert te bepalen of iemand geschikt is voor een studie aan de universiteit. Daartoe moeten enkele vragen beantwoord worden, zo moet de kandidaat weten op welke dag hij/zij geboren is. En als je niets van vermenigvuldigen weet, is een beta-studie misschien niet verstandig.

Om te beginnen moet de gebruiker diens geboortejaar als geheel getal invoeren, en daarna de geboortemaand, ook als geheel getal. Vervolgens voert men de geboortedag in, wederom als geheel getal. Het programma berekent dan de leeftijd, zowel in aantal jaren als in maanden (bijvoorbeeld: 10 jaar en 3 maanden; 123 maanden); beide worden op het beeldscherm getoond. De leeftijd in maanden wordt analoog aan die in jaren bepaald (als je op de 31ste geboren bent, wordt je iedere maand een maand ouder, maar je bent niet zo vaak "maandig" — dat ben je namelijk alleen op iedere 31ste). Jarige en maandige gebruikers worden gefeliciteerd. Aangenomen mag worden dat het programma op de peildatum *25 september 2023* draait (gebruik `const`; **liefhebbers** mogen met `ctime` de echte huidige dag opvragen en gebruiken). Let op: het programma moet in principe ook op andere peildata vanaf heden tot 2100 correct werken!
Gebruikers jonger dan 10 jaar (de 10-de verjaardag nog niet gevierd) of ouder dan 100 jaar (dus 101-ste verjaardag reeds gevierd) worden meteen geweigerd. Als uit het geboortejaar direct al duidelijk is dat het met de leeftijd niets gaat worden, hoeven de vragen naar maand en/of dag niet gesteld te worden. Maar soms biedt pas de dag uitsluitsel!
Nu moet de gebruiker diens geboortedag (zondag, maandag, ..., zaterdag) weten. Als deze fout is, wordt men meteen "verwijderd", en stopt het programma. Het antwoord moet met *één letter* (de eerste letter van de dag; geen cijfer dus) worden gegeven, bijvoorbeeld w voor woensdag. In het geval van d/z wordt nog om de tweede letter gevraagd.
Het is *niet* de bedoeling `ctime` te gebruiken om deze dag uit te rekenen. Het programma moet een zelf bedachte berekening bevatten om deze dag te bepalen! Gebruik bijvoorbeeld dat 1 januari 1901 op een dinsdag viel. Gebruik *niet* het **Doomsday algoritme** (zie ook **hier**), en ook *niet* allerlei ingewikkelde formules. Voor de periode 1901–2099 geldt dat een jaar een schrikkeljaar is precies dan als het jaartal door 4 deelbaar is.

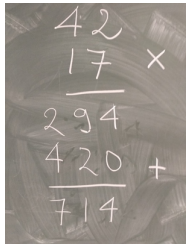
De echte test bestaat uit enkele vragen. Mensen van 30 jaar of ouder worden hierbij minstens twee maal "netter" aangesproken dan jongeren. Splits de C++-code in het programma niet onnodig vaak!

Er wordt gekeken of de aanstaande student enigszins kan vermenigvuldigen. Wiskundig inzicht is namelijk vereist voor een beta-studie. Mocht dat niet zo zijn, wordt er getest hoe het met de kunst- of literatuurkennis staat.

De gebruiker moet allereerst het product van twee willekeurige gegeven positieve gehele getallen schatten, zeg 42 en 17. Beide getallen moeten tussen 10 en 99 liggen, grenzen inbegrepen. De gebruiker krijgt, of diens antwoord goed of fout is, de bekende vermenigvuldiging te zien. In ons voorbeeld:

```
  42
  17 x
  --
 294
 420 +
 ---
 714
```

De getallen en streepjes moeten verticaal goed uitgelijnd staan. Als een rij 0 zou worden (bijvoorbeeld bij 42 maal 20) bestaat het antwoord alleen uit de andere rij: de optelling hoeft dan niet. De tweede rij streepjes (als die er is) heeft drie of vier streepjes, afhankelijk van de lengte van het product.



De *verhouding* tussen het gegokte en het juiste antwoord mag hoogstens `EPSILON` (een zekere waarde, een double, bijvoorbeeld 0.1) van 1 verschillen. Anders wordt het fout gerekend. In het voorbeeld (42 maal 17, foutmarge 0.1) moet het antwoord tussen 643 en 785 liggen, grenzen inbegrepen.

Voor het fabriceren van willekeurige gehele getallen moet gebruik worden gemaakt van de random-generator uit C++. Gebruik bijvoorbeeld `x = rand ( ) % 20;` om een "willekeurig" getal tussen 0 en 19 (grenzen inbegrepen) in de `int` variabele `x` te krijgen. Zet bovenaan in main: `srand (42);` of `srand (jaar);` (nadat `jaar` een waarde heeft gekregen), om de random-generator eenmalig te initialiseren. In plaats van 42 mag ook een ander getal staan — of zelfs, voor liefhebbers, de tijd. En soms is hiervoor `#include <cstdlib>` nodig, helemaal bovenaan het programma.

Is het antwoord goed, dan wordt de kandidaat tot een exacte studie toegelaten, en stopt het programma. Anders wordt één meerkeuzevraag (Aa/Bb/Cc/Dd) over kunst of literatuur gesteld, die uitsluitsel biedt over de toelating tot een alpha-studie. Als het daar ook mis gaat, is men helaas niet geschikt voor een universitaire studie. Gebruikers tot of tot en met (kies zelf) 30 jaar krijgen hier een andere vraag dan de oudere gebruikers — maar bij beiden is "hetzelfde" antwoord, bijvoorbeeld steeds B, goed. Wederom, of het antwoord goed of fout is, het juiste antwoord wordt steeds op het scherm afgedrukt.

Opmerkingen

Als de gebruiker een niet bestaande maand invoert, bijvoorbeeld −8, of een jaartal als 4242 (in de toekomst dus), stopt het programma met de mededeling dat dit niet kan (gebruik `return 1;`). Evenzo voor een niet bestaande dag, bijvoorbeeld 31 april of 42 december. We nemen aan dat de gebruiker zo vriendelijk is verder geen fouten te maken bij het invoeren van gegevens: men voert niet al te gekke getallen of letters in, etcetera. Vanzelfsprekend worden wel duidelijke vragen gepresenteerd.

Elk programma moet bij het "runnen" aan het begin op het beeldscherm laten zien wie de makers zijn, wat hun jaar van aankomst, studierichting en studentnummer is, welke opgave het is, wat de gebruiker te wachten en te doen staat, de datum waarop het programma gemaakt is, enzovoorts. Dit noemen we het **infoblokje**. Probeer dit er netjes uit te laten zien. Maak geen al te complexe kaders eromheen; gebruik liefst alleen de eerste 128 gewone karakters.
Bovenaan het programma (in de C++-code dus) staat uiteraard commentaar, waarin een aantal van deze elementen ook weer terugkomen, maar dan meer gericht op programmeurs, bijvoorbeeld de naam van de gebruikte compiler.

Denk aan het gebruik van lege regels, inspringen, commentaar, constanten, enzovoorts. Bovenaan het programma dient zoals gezegd commentaar over het programma te staan, speciaal bestemd voor andere *programmeurs* (en nakijkers), bijvoorbeeld kort wat het programma doet, en welke compiler gebruikt is: gebruikers van het programma vinden dat laatste niet interessant. Het infoblokje moet tijdens het "runnen" van het programma op het scherm komen, en is bestemd voor *gebruikers* van het programma. Lees ook eens over **richtlijnen** bij het maken van programmeeropgaven, en bestudeer de **huisregels**. Er hoeft geen gebruik van functies, arrays en het while- en for-statement gemaakt te worden. Alleen de headerfiles `iostream` mag en moet gebruikt worden — en eventueel `ctime` voor liefhebbers; en misschien `cstdlib` voor het gebruik van de random-generator. Ruwe indicatie voor de lengte van het C++-programma: 200 regels (300 mag ook wel). Letters moeten als `char` worden ingelezen, dus *niet* met strings, die mogen namelijk niet gebruikt worden.

Uiterste inleverdatum: **maandag 25 september 2023, 18:00 uur**.
De manier van inleveren (één exemplaar per koppel, dat — ter herinnering — uit maximaal twee personen bestaat) is als volgt.

- Digitaal de C++-code inleveren via Brightspace > Course Tools > Assignments. Stuur geen executable's, LaTeX-files of PDF-files, lever alleen één C++-file digitaal in!i
- Doe een print van het verslag in de doos bij kamer 159 van het Snellius.

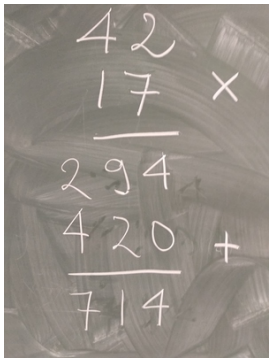
De laatst voor de deadline ingeleverde versie wordt nagekeken.

Overal duidelijk datum en namen van de twee makers vermelden, in het bijzonder als commentaar in de eerste regels van de C++-code. Lees bij het **derde werkcollege** hoe het verslag eruit moet zien. Zijn spaties/tabs goed verwerkt?

www.liacs.leidenuniv.nl/~kosterswa/pm/op1pm.php

Voor de eerste programmeeropgave moet je onder meer, voor een gegeven datum, de dag van de week uitrekenen. Bijvoorbeeld: 1–1–1901 $\longrightarrow$ dinsdag.

Hoe zou je dat uit je hoofd doen?



Wanneer is een jaar een schrikkeljaar?

www.liacs.leidenuniv.nl/~kosterswa/pm/op1pm.php

Verslag in LaTeX! En wat is 42×17?

Niet gebruiken: Doomsday-algoritme 5–9: dinsdag
4–4, 6–6, 8–8, 10–10, 12–12, "I work from 9–5 in a 7–11".

In een Linux windows-omgeving zoals WSL start je een of meer terminals: windows waarin je tekst-georiënteerde opdrachten kunt geven. Daarin tik je in:

`gedit een.cc &`	edit je eerste C++-programma; open hiertoe een "edit-window"
`g++ -Wall -o een een.cc`	compileer `een.cc` naar `een`
`./een`	run de executable `een`
`ls -lrt`	overzicht van je files (%)

De ampersand `&` zorgt er voor dat je in het oorspronkelijke window ook kunt doorwerken. En (%) levert zoiets als:

```
-rw-r--r-- 1 kosterswa domain users   124 Sep 7 12:42 een.cc
-rwx------ 1 kosterswa domain users 11049 Sep 7 12:43 een
```

Eerste werkcollege

- donderdag 13:15–. . . (Informatici)
- met aansluitend vragenuur
- vrijdag 11:00–12:45 (Wiskundigen; niet in 302 en 306)
- met aansluitend vragenuurtje
- op eigen laptop in computerzalen Snellius
- www.liacs.leidenuniv.nl/~kosterswa/pm/pmwc1.php
- doel: compiler, Hello world, int, opgave, . . .

- Hello world voor C++
- boek en “dictaat”
- ULCN-account
- software voor thuis/laptop: WSL, . . .
- de eerste programmeeropgave
- www.liacs.leidenuniv.nl/~kosterswa/pm/
- huiswerk: Savitch Hoofdstuk 1; dictaat 1 en 2