
C++ifPCwhileLinuxdiffintFirefoxboolpublicprivateWindows
fileg++UNIXVisualOSFedoraIscharpsopenclosegetputarray
XPcpelmmuttcincoutmainRedHatmkdireditorenumDebian
classusingmathfloattruecpSUSEnamespaceKnoppixinclude
doublecdGNUdokitobjectcompilermorelinkiostreamelsenot
WWWmakefilegrepiomanipgeditGnomestructfstreamrmfalse
andorDev-C++topforstringemailmanlprchmodKDEKubuntu

Programmeermethoden

Introductie

Walter Kusters

week 1: 5–9 september 2011

<http://www.liacs.nl/home/kusters/pm/>

Het college wordt gegeven door dr. W.A. (Walter) Kusters.

Het werkcollege wordt verzorgd door

- Iris Hupkens,
- Jan Kalmeijer,
- Jan van Rijn (Den Haag),
- Bas van Stein,
- Marijn Swenne en
- Edwin Veger (Den Haag),

onder leiding van

- Frank Takes.

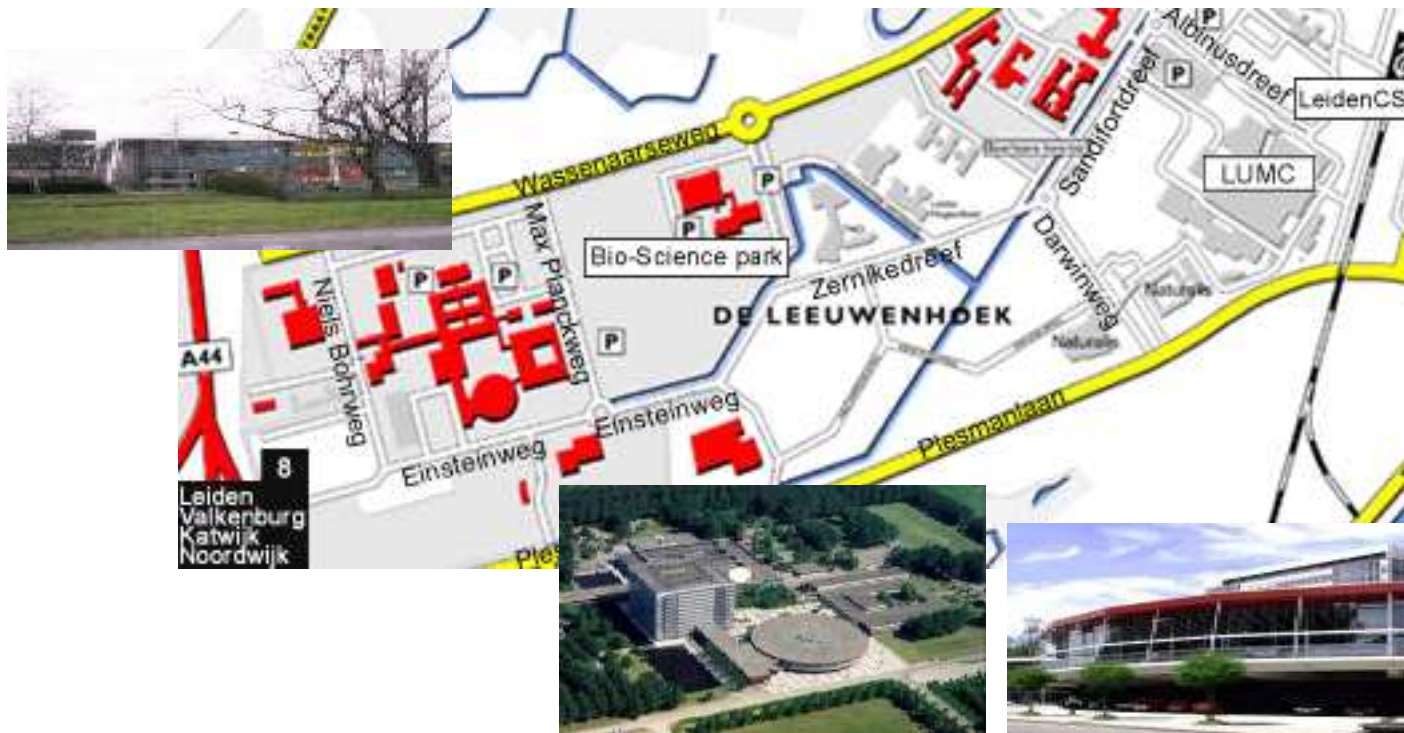
De **colleges** zijn op dinsdagen, 9.00–10.45 uur. Plaats: Gorlaeus, zaal 3 (op 8–11, 22-11 en 29–11 in zaal 2). Eerste keer: 6–9–2011; laatste keer: 13–12. Op 25–10 is er geen college.

De **werkcolleges** zijn voor Wiskundigen en Informatici op dinsdagen, 11.15–13.00 uur, en voor Natuur- en Sterrenkundigen op donderdagen, 11.15–13.00 uur. In de week van 24–10 zijn er geen werkcolleges.

Plaats: computerzalen 302/304 en 306/308 (PCs, Linux en Windows), 307 en 309 (PCs, Linux; wiskundigen) van het gebouw van Wiskunde en Informatica: het Snellius.

Er geldt een **aanwezigheidscontrole!**

Snellius



Gorlaeus

De Haagse **colleges** zijn op *maandagen*, 11.15–13.00 uur. Plaats: dertiende etage station Den Haag CS, wisselende zaal. Eerste keer: *donderdag* 8–9–2011, 13.45–15.30 uur; laatste keer: 12–12. En niet op 3–10.



De **werkcolleges** zijn op maandagen, 13.45–15.30 uur; niet op 3–10. Eerste keer: donderdag 8–9–2011, 15.45–17.30 uur. Plaats: computerzaal “Paleistuin”, 13.25.

Er geldt een **aanwezigheidscontrole!**

Hoe zit het met al die “accounts”?

- *alle studenten*: ULCN-account
A.Einstein@umail.leidenuniv.nl
- *Informatici, Natuur/Sterrenkundigen*:
UNIX-account Snellius 302/304 en 306/308
aeinstei@liacs.nl
- *Informatici*: Windows-account Snellius 302/304 en 306/308
aeinstei@liacs.nl
- *Wiskundigen*: UNIX-account Snellius 307, 309
aeinstei@math.leidenuniv.nl
- *Natuur/Sterrenkundigen*: Windows-account Huygens

Het **schriftelijk tentamen** is op

- woensdag 4 januari 2012, 10.00–13.00 uur
vragenuur: dinsdag 3 januari 2012, 11.00 uur, zaal 174,
Snellius
- hertentamen:
woensdag 28 maart 2012, 14.00–17.00 uur
- hertentamen:
maandag 30 juli 2012, 14.00–17.00 uur

Plaats: Snellius-gebouw.

NB Het januari-tentamen is ook in Den Haag.

Zie de website voor oude tentamens en uitwerkingen.

Er moeten vier **programmeeropgaven** gemaakt worden. Als deze alle voldoende zijn (hooguit één vijf mag) *en* het tentamen voldoende 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!

- in tweetallen (of individueel, dus ≤ 2) 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
- middag-spreek/vragenuur: 15.30–17.00 uur, in computerzalen (di/wo/do en deadline-vr)

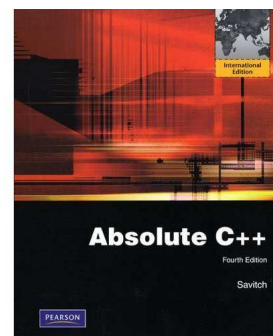
- Opgave 1: **vrijdag 23 september 2011**, 17.00 uur
- Opgave 2: vrijdag 14 oktober 2011
- Opgave 3: vrijdag 18 november 2011
- Opgave 4: vrijdag 9 december 2011

In te leveren: geprinte versie met verslag (doos in kamer 156, Snellius) *en* digitaal de C++-code naar `pm@liacs.nl`.
Deadlines Den Haag: maandagen 26-9-2011, 24-10, 21-11, 12-12, 11.00 uur; geprinte versie bij docent.

We maken gebruik van het volgende boek:

W. Savitch
Absolute C++
fourth edition
Addison-Wesley, 2010

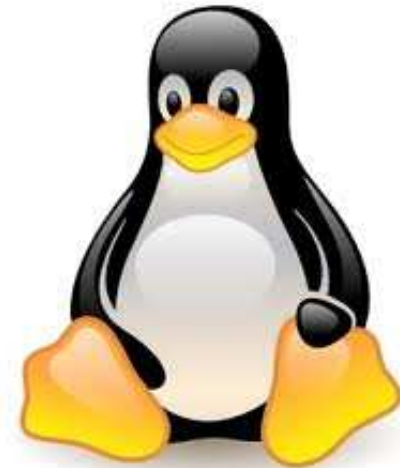
Oudere drukken zijn ook goed (ouderejaars!). En er zijn vele andere boeken.



En het “dictaat”: zelf printen.

Naast **Microsoft Windows** kunnen PCs ook draaien onder **Linux**, een **operating systeem** (OS) uit de UNIX-wereld.

Met een Kubuntu Live-CD of -DVD start je PC meteen op in **KDE**, de K Desktop Environment, een grafische windows-omgeving bovenop Linux. Let er op dat je op een verstandige plaats (USB) moet saven.



Je kunt ook “gratis” distributies als SUSE, RedHat/Fedora, Debian, ... gebruiken — maar dan moet je er iets meer vanaf weten.

week	onderwerp	boek	dictaat
5–9 sep	Introductie: UNIX, C++, machines	1	1,2
12–16 sep	Types		3.1/3,3.9,op1/5
19–23 sept	Controlestructuren	2	3.4/5,op6/10
26–30 sep	Functies, files	3,4, 12.1/2	3.6/7,op11/17
4–7 okt (⊠)	OOP	6,7.1	3.11,op18/25
10–14 okt (⊠)	Vervolg		4.1,op26/30
17–21 okt (⊠)	Arrays	5	3.8,op31/36
...			

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

In rood: de weken met een deadline op vrijdag.

(⊠) In Den Haag een week later.

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”: `inklude`.
3. Draai (= run) deze **executable** vanuit het OS.
Eventuele fouten: “run-time-fouten”: deel door 0.

Herhaal deze cyclus zo vaak als nodig.

Soms (Dev-C++) is een en ander geïntegreerd.

Een eerste C++-programma:

```
#include <iostream>
using namespace std;
int main ( ) {
    cout << "Dit komt op het scherm." << endl;
    return 0;
} //main
```

Dit programma zet alleen een tekstje op het beeldscherm.
(In Dev-C++ zie je soms niet veel.)

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

Een tweede C++-programma:

```
// dit is een simpel programma
#include <iostream>
using namespace std;
int main ( ) {
    int getal = 42; // een variabele declareren
                  // en initialiseren
    cout << "Geef een geheel getal .. " << endl;
    cin >> getal;
    cout << "Kwadraat is: "
         << getal * getal << endl;
    return 0;
} //main
```



Programmeermethoden 2011 Eerste programmeeropgave: Da Vinci

De eerste programmeeropgave van het vak **Programmeermethoden** in het najaar van 2011 heet *Da Vinci*; zie ook het eerste, tweede en derde werkcollege.
Spreek/Vragenuur in zalen 302 .. 309: dinsdag 6, woensdag 7, donderdag 8, dinsdag 13, woensdag 14, donderdag 15, dinsdag 20, woensdag 21, donderdag 22 en vrijdag 23 september 2011, van circa 15.30 tot 17.00 uur.

Het is de bedoeling om een C++-programma te schrijven dat beoordeelt of de gebruiker knap genoeg is om een code te raden. Allereerst moet het programma iemands geboortedag en leeftijd bepalen. Daarna worden twee lastige vragen gesteld.

De opgave

Om te beginnen moet de gebruiker zijn/haar geboortjaar als geheel getal invoeren, en daarna de geboortemaand, ook als geheel getal. Vervolgens voert hij/zij de geboortedag in, wederom als geheel getal. Het programma berekent dan de leeftijd van de gebruiker, 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). Aangenomen mag worden dat het programma op de peildatum **23 september 2011** 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 geboortjaar direct al duidelijk is dat het met de leeftijd niets gaat worden, hoeven de vragen naar maand en dag niet gesteld te worden. Maar soms biedt pas de dag uitsluit!

Nu moet de gebruiker, als verificatie, zijn/haar geboortedag (zondag, maandag, ..., zaterdag) weten. Als deze fout is, wordt men meteen "verwijderd", en stopt het programma. Het antwoord moet met één letter worden gegeven, bijvoorbeeld w voor woensdag. Bedenk zelf iets voor het d- en z-probleem. Het is *niet* de bedoeling `ctime` te gebruiken om deze dag uit te rekenen. Het programma moet een 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). Voor de periode 1901–2099 geldt dat een jaar een schrikkeljaar is precies dan als het door 4 deelbaar is.

De echte test bestaat uit een tweetal vragen. Mensen van 18 jaar of ouder worden bij iedere vraag met "u" aangesproken, jongeren met "je/jij". Splits de C++-code in het programma niet onnodig vaak!

Allereerst moet de gebruiker een meerkeuzevraag (A/B/C/D) goed beantwoorden. Is het antwoord fout, dan stopt het programma. Gebruikers tot en met 30 jaar krijgen een andere vraag dan de oudere gebruikers. Verder is de vraag dus steeds hetzelfde. De twee vragen moeten over *kunst* gaan. Hoofdletters en kleine letters moeten natuurlijk goed afgehandeld worden.

Als laatste moet de gebruiker twee willekeurige gegeven gehele getallen tussen 1 en 999 op elkaar delen. Gebruik hierbij bijvoorbeeld `x = rand () % 1000`; om een "willekeurig" getal tussen 0 en 999 in de variabele `x` te krijgen.

De gebruiker moet het antwoord twee maal geven: als geheel getal (dat precies goed moet zijn) en als decimaal getal, met een decimale punt, dat er maximaal een constante `MAXFOUT` (zeg 0.1) naast mag zitten. Converteer zelf op de juiste manier tussen `int` en `double`.

Op sommige systemen moet `#include <cstdlib>` bovenaan het programma staan om `random` te mogen gebruiken. En de `random`generator kun je met `srand (123); ofsrand (jaar); "seed"-en`.

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: hij/zij voert niet al te gekke getallen of letters in, etcetera. Vanzelfsprekend worden hem/haar wel duidelijkke 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 een beetje aardig uit te laten zien. Maak geen at te complexe kaders omhoog; gebruik liefst alleen de eerste 128 gewone karakters.

Bovenaan het programma (in de C++-code dus) staat ultraard 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, commentaar, constanten, enzovoorts. Bovenaan het programma dient ook 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 headerfile `iostream` mag en moet gebruikt worden — en eventueel `ctime` en `string` voor liefhebbers; en misschien `cstdlib`. Ruwe indicatie voor de lengte van het C++-programma: 200 regels (300 mag ook wel).

Uiterste inleverdatum: **vrijdag 23 september 2011, 17.00 uur**.

 Haagse studenten: maandagochtend 26 september 2011, 11.00 uur (voor het college).


Manier van inleveren (één exemplaar per koppel, dat — ter herinnering — uit maximaal twee personen bestaat):

1. Digitaal de C++-code inleveren: stuur een email naar `pm@liacs.nl`. Stuur geen executables, lever alleen de C++-file digitaal in! Noem deze bij voorkeur zo iets als `obamabiden1.cc`, dit voor de eerste opdracht van het duo Obama-Biden. De laatste voor de deadline ingeleverde versie wordt nagekeken.
2. En ook een papieren versie van het verslag (inclusief de C++-code) deponeren in de speciaal daarvoor bestemde doos "Programmeermethoden" in de postkamer van Informatica, kamer 156 van het Snellius-gebouw.

 Haagse studenten: bij de docent, tijdens een (werk)college.

Overal duidelijk datum en namen van de (maximaal 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?

Te gebruiken compiler: als hij maar C++ vertaalt; het programma moet in principe zowel op een Linux-machine (met `g++`) als onder Visual C++ of Dev-C++ draaien. Test dus in principe op beide systemen! Normering: (consequente) layout 2; commentaar 2; infoblokje 2; werking 4. Eventuele aanvullingen en verbeteringen: lees deze WWW-bladzijde: www.liacs.nl/home/kosters/pm/op1pm.php.

 Tim Cox & Walter Kosters

Voor de eerste programmeeropgave moet je onder andere, 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?

<http://www.liacs.nl/home/kosters/pm/op1pm.php>

Verslag in L^AT_EX!

Doomsday:

4-4, 6-6, 8-8, 10-10, 12-12, "I work from 9-5 in a 7-11".

```
// 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 ( ) {
    double straal; // straal van de cirkel
    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
```

```
// 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 ( ) {
    double straal; // straal van de cirkel
    cout << "Geef straal, daarna Enter .. ";
    cin >> straal;
    if ( straal > 0 ) { // accolades nodig!
        cout << "Oppervlakte ";
        cout << pie * straal * straal << endl;
    }//if
    else
        cout << "Niet zo negatief ..." << endl;
    cout << "Einde van dit programma." << endl;
    return 0;
}//main
```

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

```
gedit een.cc &
```

```
g++ -Wall -o een een.cc  
./een
```

```
ls -lrt
```

edit je eerste C++-programma;
open hiertoe een “edit-window”
compileer een.cc naar een
run de executable een
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 kosters users   124 2011-09-06 10:39 een.cc  
-rwx----- 1 kosters users 11049 2011-09-06 10:40 een
```

- in computerzalen
- `http://www.liacs.nl/home/kosters/pm/pmwc1.php`
- doel: UNIX, Hello world
- UNIX-account; bij inloggen: **selecteer KDE**

- Hello world
- boek en “dictaat”
- diverse paswoorden = wachtwoorden: werkcollege
- software voor thuis
- de eerste programmeeropgave
- `http://www.liacs.nl/home/kosters/pm/`
- huiswerk: Savitch Hoofdstuk 1; dictaat 1 en 2