



HOGESCHOOL ROTTERDAM / CMI

Programmeren 1: Imperatief Programmeren

TINPRO02-1

Aantal studiepunten: 2

Cursusbeheerder: dr. ir. M. Hajian & dr. W. M. Bergmann Tiest



Cursusbeschrijving

Cursusnaam:	Programmeren 1: Imperatief Programmeren
Cursuscode:	TINPRO02-1
Aantal studiepunten en studiebelastinguren:	2 ec Dit studieonderdeel levert de student 2 ec op, hetgeen overeenkomt met een studielast van 56 uren waarvan 16 contacturen. De verdeling van deze 56 uren is als volgt: <ul style="list-style-type: none"> • 7 × 2 uur gecombineerd hoor- en werkcollege: 14 uur. • mondelinge toelichting inleveropdrachten 2 uur. • onbegeleid werken aan werkcollege en inleveropdrachten: 40 uur.
Vereiste voorkennis:	Geen.
Werkvorm:	Gecombineerd hoor- en werkcollege. Tijdens en buiten de contacturen wordt gewerkt aan werkcollege- en inleveropdrachten. Aftekenen van de werkcollege-opdrachten is verplicht. Tijdens de laatste les moeten studenten kort hun inleveropgaven mondeling toelichten.
Toetsing:	De toetsing bestaat uit een drietal verplichte inleveropgaven, die beoordeeld worden en samen het eindcijfer vormen.
Leermiddelen:	Reader: https://www.tutorialspoint.com/cprogramming . Slides worden op Google Classroom geplaatst voor de start van de les. Studenten worden geacht in het bezit te zijn van een Arduino ontwikkelbordje.
Leerdoelen:	De student: <ol style="list-style-type: none"> 1. kan een ontwerp opstellen voor een programma. 2. kan een testplan opstellen en zinnige testdata formuleren. 3. kan programma's schrijven in de programmeertaal C. 4. begrijpt hoe variabelen, verschillende datatypen, en operators werken en kan deze implementeren in C. 5. begrijpt de concepten van beslissing en herhaling en kan deze implementeren in C. 6. begrijpt de concepten van arrays, pointers, en strings en kan deze implementeren in C. 7. begrijpt de werking van functies en kan deze implementeren in C. 8. kan eenvoudige console- en file-input/output gebruiken.
Inhoud:	Tijdens deze cursus leren studenten imperatief programmeren in C om een probleem op te lossen. Het ontwerp van een programma wordt in detail behandeld om een complex probleem als een simpele grafische voorstelling weer te geven. Testen en test data worden besproken waarbij individuele eenheden/componenten van een programma getest kunnen worden (unit testing). Variabelen, verschillende datatypen, en operators worden behandeld. Vervolgens wordt de werking van beslissings- en herhalingsconstructies behandeld. Arrays, pointers, en strings worden doorgenomen. Functies, files, en I/O worden besproken. Deze kennis moet vervolgens toegepast worden in de opdrachten en werkcolleges.
Opmerkingen:	De werkcollege, en opdrachten wordt individueel uitgevoerd.
Cursusbeheerder:	dr. ir. M. Hajian & dr. W. M. Bergmann Tiest
Datum:	12 juli 2018

1 Algemene omschrijving

1.1 Inleiding

De programmeertaal C wordt veel gebruikt in embedded systemen waarbij performance belangrijk is. Omdat men deze programmeertaal dicht op de hardware zit, verschaft kennis van C inzicht in de werking van de computer. De programmeertaal C is de voorloper van moderne programmeertalen zoals Java en C++. Imperatief programmeren in C staat aan de basis van geavanceerdere programmeerparadigma's zoals object-geïntendeerd programmeren.

1.2 Relatie met andere onderwijseenheden

TINPRO02-1 bereidt studenten voor op de rest van de leerlijn programmeren, projecten 1 t/m 8, en de meeste andere TI-vakken, TINLAB, stage, en afstuderen.

1.3 Leermiddelen

Voor het uitvoeren van het werkcollege en de inleveropdrachten kan gebruik gemaakt worden van:

- de gcc compiler. gcc is een open source compiler-pakket dat vele talen ondersteunt, zoals C/C++, zie <https://gcc.gnu.org/>. Er kan gebruik gemaakt worden van een code editor als Visual Studio Code, zie <https://code.visualstudio.com>.
- Code::Blocks: Code::Blocks is een krachtige open source C, C++ en Fortran IDE. Het debuggen met behulp van Code::Blocks is zeer gebruiksvriendelijk. Fouten zoeken door gebruik te maken van break points is een zeer krachtig gereedschap van Code::Blocks.
- boek: *The C programming Language* door Brian Kernighan en Dennis Ritchie. (Niet verplicht.)

Verder kan gebruik gemaakt worden van websites zoals: <https://www.tutorialspoint.com/cprogramming>.

1.4 Aanwezigheid

Aanwezigheid tijdens de lessen is verplicht vanwege het laten aftekenen van werkcollege-opdrachten bij de docent. Er wordt een coulance-regeling gehanteerd waarbij maximaal 20 % van de lessen gemist mag worden, mits met goede redenen en van tevoren gemeld bij de docent.

2 Programma

Tijdens deze cursus leren studenten imperatief programmeren in C om een probleem op te lossen. Het ontwerp van een programma wordt in detail behandeld om een complex probleem als een simpele grafische voorstelling weer te geven. Testen en test data worden besproken waarbij individuele eenheden/componenten van een programma getest kunnen worden (unit testing). Variabelen, verschillende datatypen, en operators worden behandeld. Vervolgens wordt de werking van beslissings- en herhalingsconstructies behandeld. Arrays, pointers, en strings worden doorgenomen. Functies, files, en I/O worden besproken. Deze kennis moet vervolgens toegepast worden in de opdrachten en werkcolleges.



Week	Lesinhoud
2	Introductie: editor en compiler. flowchart, variabelen, verschillende datatypen.
3	Operatoren
4	Debugging, test plan en unit test, test data. Inleveropdracht 1.
5	Beslissing: if...else, if...elseif...else, switch...case.
6	Herhaling: for loop, while loop, do...while loop. Inleveropdracht 2.
7	Functies. Console- en file-input/output.
8	Strings arrays, en pointers. Inleveropdracht 3.
9	Mondelinge toelichting van inleveropdrachten.

3 Toetsing

3.1 Procedure

De toetsing bestaat uit een drietal verplichte inleveropgaven, die beoordeeld worden en samen het eindcijfer vormen. Tevens moeten de werkcollegeopdrachten afgetekend zijn om in aanmerking voor een eindcijfer te komen. De werkcollegeopdrachten hebben echter geen effect op de hoogte van het eindcijfer. Werkcollege-opdrachten moeten uitgevoerd worden tijdens de betreffende les.

De inleveropdrachten moeten vergezeld worden van een ontwerp (bijv. flowchart) en een kort testplan en testrapport. Tevens moet de code voorzien zijn van informatief commentaar.

Deadline: de inleveropdrachten moeten één week na het verstrekken van de opdracht, uiterlijk om 17:00 uur ingeleverd worden.

Mondelinge toelichting: de ingeleverde opdrachten moeten tijdens de laatste les mondeling toegelicht worden, waarbij de student vragen over de werking van zijn/haar code moet beantwoorden. Deze antwoorden wegen mee in de beoordeling.

Beoordeling: Voor ieder van de drie inleveropdrachten geldt de volgende beoordeling:

De opdracht voldoet niet aan de gestelde eisen	0 pnt.
Het ontwerp en testplan zijn in orde	1 pnt.
De opdracht voldoet aan de gestelde eisen	1 pnt.
De opdracht voldoet aan de gestelde eisen en bevat extra functionaliteit zoals aangegeven in de opdrachtomschrijving	1 pnt.

Eén extra punt kan toegekend worden indien de drie inleveropdrachten consistent een grote technische complexiteit of een zeer elegante oplossing laten zien.

De punten voor de drie inleveropdrachten en het eventuele bonuspunt worden bij elkaar opgeteld en vormen het eindcijfer voor de cursus. De voorwaarde voor verkrijgen van een eindcijfer is dat voor alle drie de opdrachten minstens twee punten behaald zijn en het werkcollege afgetekend is.

3.2 Herkansing

De herkansing bestaat uit het opnieuw inleveren van de inleveropdrachten in week 6 van OP2.

Hogerejaars studenten die het vak nog moeten herkansen kunnen naar keuze de opdrachten inleveren of deelnemen aan een schriftelijke toets over de inhoud van het vak van vorig jaar. Meld je hiervoor bij de cursusbeheerder.

Bijlage: Toetsmatrijs

De opdrachten zijn gekoppeld aan de leerdoelen volgens onderstaande matrix:

- Opdracht 1 \Rightarrow Leerdoel 1, 2, 3, 4.
- Opdracht 2 \Rightarrow Leerdoelen 1, 2, 3, 4, 5.
- Opdracht 3 \Rightarrow Alle leerdoelen.