

HBO ICT

hoofdfase

HBO ICT

TI

*Studenten*handleiding

**C++ Programming & Software Engineering 1**

**TICT-V1OOPC-16**

Studiejaar 2017-2018

|  |  |
| --- | --- |
|  |  |
| Cursuseigenaar | Wouter van Ooijen |
| **Auteur(s)** | Wouter van Ooijen |
| **Datum** | 2017-09-04 |
| Versie | 2.0 |
|  | |

© Institute for ICT, Hogeschool Utrecht, 2017

# Inhoudsopgave

[Inhoudsopgave 2](#_Toc474153316)

[1 Opzet cursus 3](#_Toc474153317)

[1.1 Inleiding 3](#_Toc474153318)

[1.2 Praktijkvoorbeeld 3](#_Toc474153319)

[1.3 Plaats cursus binnen onderwijsprogramma 3](#_Toc474153320)

[1.4 Inhoud 3](#_Toc474153321)

[1.4.1 Beroepstaken en professional skills 3](#_Toc474153322)

[1.4.2 Competentiematrix 3](#_Toc474153323)

[1.4.3 Leerdoelen 4](#_Toc474153324)

[1.4.4 Kennisbasis 4](#_Toc474153325)

[1.5 Toetsing 4](#_Toc474153326)

[1.5.1 Opzet 4](#_Toc474153327)

[1.5.2 Practica 5](#_Toc474153328)

[1.5.3 Tentamen 5](#_Toc474153329)

[1.5.4 Excellentiemogelijkheden 5](#_Toc474153330)

[1.6 Leeromgeving 5](#_Toc474153331)

[1.6.1 Opzet: werkvormen, type bijeenkomsten, opdrachten, begeleiding 5](#_Toc474153332)

[1.6.2 Materialen 5](#_Toc474153333)

[2 Overzicht cursusweken 7](#_Toc474153334)

[3 Gegevens studiegids 9](#_Toc474153335)

# Opzet cursus

## Inleiding

In dit vak leert de student technieken voor het ontwikkelen van software voor een klein embedded systeem. Hierbij wordt aandacht besteed aan het bouwproces (preprocessen, compileren, linken, make, separate compilatie), een aantal C++ aspecten (templates, lambda’s, constexpr) en assembler.

## Praktijkvoorbeeld

|  |  |  |
| --- | --- | --- |
|  |  |  |

De manier van programmeren is passend voor het ontwikkelen van kleine (micro-controller) devices.

## Plaats cursus binnen onderwijsprogramma

Het vak bouwt voort op de programmeervakken die in de propedeuse gegeven worden:

* Programming
* Algorithms and Datastructures in C
* Object Oriented Programming in C++
* Individueel Propedeuse Assessment

Parallel met dit vak wordt het vak Modelleren gegeven, waarin de student leert met behulp van een (subset van) UML een system te ontwerpen. Het RTOS wordt in beide vakken gebruikt.

Aansluitend gaat de student in een thema-opdracht de aangeleerde ontwerp- en programmeervaardigheden toepassen om een real-time systeem (Laser Tag) te ontwerpen en programmeren.

## Inhoud

### Beroepstaken en professional skills

In deze module wordt gewerkt aan algemene programmeervaardigheden en meer specifieke programeervaardigheden voor kleine embedded systemen.

### Leerdoelen

De student kan:

* een make-based application building systeem gebruiken, begrijpen, en er kleine aanpassingen in aanbrengen.
* een kort (tot ~ 20 regels) en eenvoudig (geen variabelen op de stack) stukje assembler schrijven in een specifieke assemblertaal.
* vanuit assembler C/C++ code aan roepen, en vanuit C/C++ code assembler aan roepen, inclusief eenvoudge parameteroverdracht (in registers)
* een kort (tot ~20 regels) stukje assembler doorgronden en verbeteringen in aanbrengen.
* een geautomatiseerde unit-test maken met behulp van een eenvoudige unit test library.
* de concurrency mechanismen begrijpen en gebruiken met behulp van een multithreading library.
* omgaan met in de cursus beschreven C++ en S.E. aspecten.

### Kennisbasis

De steekwoorden voor de kennisbasis zijn in de volgende tabel toegelicht.

|  |  |
| --- | --- |
| **Steekwoorden** | **Onderwerpen die in de module aan de orde komen** |
| make, build systemen | De manier waarop een applicatie wordt opgebouwd uit apart gecompileerde source files; de rol van de linker hierin; de rol van make (scripts) in het bouwen van een applicatie. |
| assembler | Programmeren in Cortex-M0 assembler; combineren van C/C++ en assembler; lezen van assembler; tellen van clock cycles (performance). |
| concurrency, threading | Context switching mechanisme; synchronization mechanismen van het RTOS |
| C++ | Constexpr;Templates, SFINAE. |
| software engineering | Design patterns: proxy, RAII |

## Toetsing

### Opzet

De cursus wordt beoordeeld aan de hand van de volgende toetsen:

* De 5 practicum opgaven (alle opgaven moeten voldoende zijn, telt voor 0 %)
* Het tentamen (telt voor 100 %)

Beide toetsen dien je voldoende te maken om te slagen voor de cursus.

### Practica

Het document met de practicumopgaven staan op sharepoint. Alle opgaven moeten individueel gemaakt worden en aan de docent getoond worden ter goedkeuring. Alle opgaven moeten voldoende beoordeeld worden.

Toon je practica per week aan de docent ter goedkeuring. Je moet de opgave(n) werkend tonen, en de docent kan je code bekijken en je vragen onderdelen uit te leggen. Het nominale inlevermoment van een practicum is in de week van dat practicum. Later inleveren kan, maar dan heeft de docent waarschijnlijk minder gelegenheid om feedback te geven. De uiterste inleverdatum is een dagdeel in de tentamenperiode, die de docent in overleg met de studenten inplant. Dat dagdeel is beperkt in tijd: er is dus geen garantie dat iedereen aan de beurt komt. Zorg dus dat je eerder (in de lesweken) je practica aftekent.

De herkansingsmogelijkheid voor de practica is in het blok volgend op het blok waarin de cursus gegeven wordt. Hiervoor hoeven alleen die practica die niet als voldoende waren beoordeeld te worden getoond.

De practica zijn een aparte entry in Osiris, een voldoende voor alle practica blijft dus staan, ook als het tentamen niet voldoende is.

Als slechts een deel van de practica voldoende gemaakt is dan is het mogelijk dat in een volgend jaar een nieuwe set practica gedaan moet worden.

### Tentamen

Het tentamen bestaat uit voor 50% uit open vragen en 50% uit multiple-choice vragen. Beide onderdelen kunnen 45 van de 100 punten opleveren. 10 punten krijg je gratis, het minimum cijfer is dus 1.0.

Voor de open vragen code (C++ en/of assembler) geschreven moeten worden. Hierbij geldt dat kleine schrijffouten die door de compiler/assembler gedetecteerd zouden worden en waarvan de verbetering vanzelfsprekend is niet tot aftrek van punten leiden.

Bij het tentamen wordt een copie van de ‘assembler summary’ gegeven. Dit document is ook op sharepoint te vinden. Verder zijn bij het tentamen geen hulpmiddelen toegestaan.

### Excellentiemogelijkheden

Voor het behalen van excellentiesterren of andere uitdagingen kan de student contact opnemen met de docent.

## Leeromgeving

### Opzet: werkvormen, type bijeenkomsten, opdrachten, begeleiding

Binnen deze cursus worden verschillende bijeenkomsten georganiseerd.

* **Hoorcollege (2 \* 2 uur, per klas of met de klassen samengevoegd)**: de docent ligt de stof toe. Er is natuurlijk ook gelegenheid tot stellen van vragen.
* **Werkcollege ( 2 \* 2 uur per klas):** de student werkt aan de practicumopgaven, toont ze aan de docent ter goedkeuring, en vraagt nadere uitleg over onderwerpen die na het hoorcollege niet voldoende duidelijk zijn. Er is natuurlijk ook gelegenheid tot stellen van vragen.

### Materialen

#### Software

De documenten en code voor deze module worden beschikbaar gesteld via sharepoint. Er wordt gebruik gemaakt van de ontwikkelomgeving die ook bij IPASS is gebruikt (bmptk, hwlib, GCC voor PC en Due, Catch). Daarnaast wordtr gebruik gemaakt van een eenvoudig RTOS (Real Time Operating System).

#### Hardware

De student heeft de volgende fysieke materialen nodig:

* Een PC (windows wordt ondersteund, voor Linux zal de student zelf aan de slag moeten)
* Een Arduino Due met USB kabel, breadboard, draadjes, LEDjes, weerstanden, drukknopjes, en een 128x64 I2C Oled (dit materiaal was ook al nodig voor IPASS)
* Een luidsprekertje met weerstand (wordt door de docent ter beschikking gesteld)

#### Documenten

Op sharepoint kan je de volgende documenten vinden:

* Een (engelstalige) assembler introductie, dit is de reader voor het assembler deel.
* Een reader voor het C++ deel.
* Een document met de huiswerk opgaven.
* De powerpoint sheets
* Een document met oefenopgaven voor assember
* Een proeftentamen.

Het is de bedoeling de lessen op te nemen en deze opnamen via youtube aan de studenten ter beschikking te stellen. De link naar deze afspeellijst wordt op sharepoint geplaatst.

.

# Overzicht cursusweken

|  |  |  |  |
| --- | --- | --- | --- |
| Wk | Onderwerpen | Bestuderen | Practicum opgave |
| 1. | C++:   * Preprocessor * Compiler * Linker * Make | PP Sheets 1A  Reader C++:  2 inleiding  3 bouwen van een applicatie | Ringtone omzetter naar embedded C++ code |
| Assembler:   * Inleiding * Geschiedenis * Instructies | PP Sheets 1B  Reader Assembly:  1 Low-level computer languages  2 A bit of history  3.1 Registers  4.1 literal values  4.3 Load, store, and register-to-register move  4.4 Calculations (alleen arithmetic & logic)  4.5 Flow control instructions |
| 2. | Assembler:   * herhaling * directives * meer instructies * C/C++ & assembler interactie | PP Sheets 2A  Reader Assembly:  4.2 Directives  4.4 rest  5. Subroutines | Assembler aanroepen uit C / C++ en omgekeerd |
| C++:  Templates  Proxy pattern | PP Sheets 2B  Reader C++:  4 Templates |
| 3. | Context switching  Threads | PP Sheets 3B  Reader ‘’rtos’ | NEC IR decoder application implementeren |
| Assembler:  Nog wat instructies  Assembler listings  Startup  Linken  PP Sheets 3A  Reader assembly  H 4 nu helemaal  H6, H7 | PP Sheets 3A  Reader Assembly:  4.6 miscaleneous  6 startup  7 building an application  8 context switching |
| 4. | C++  Reader week 4  constexpr | PP Sheets 4A  Reader C++:  5 constexpr | OLED klokje zonder run-time floating-point berekeningen |
| Assembler:  pipelineing  cycle counting | PP Sheets 4B  Reader Assembly:  3.2 pipelining |
| 5. | RAII  Lempel-Ziv  Metaprogramming | PP Sheets 5  Reader C++:  7 Lempel-Ziv  8 Metaprogramming | Lempel-Ziv decompresie in C en in assembler |
| 6. | Het proeftentamen wordt besproken. | proeftentamen | (geen) |

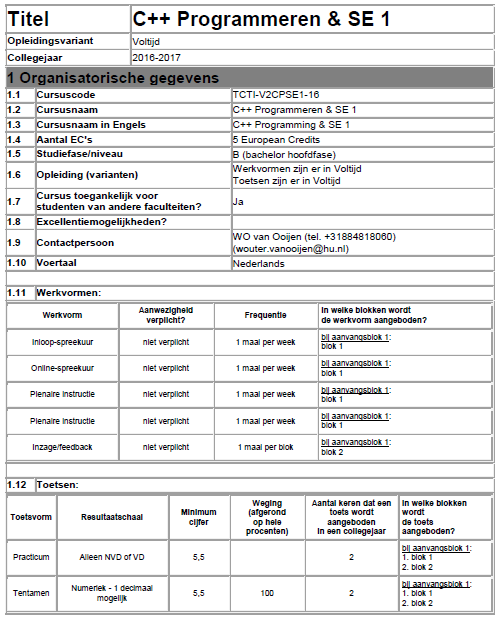
De totale stof amvat alle hoofdstukken van de beide readers (Assembler en C++). Als je dus het hele vak wil leren hoeft niet uit de tabel te halen welke delen in welke weken behandeld worden.

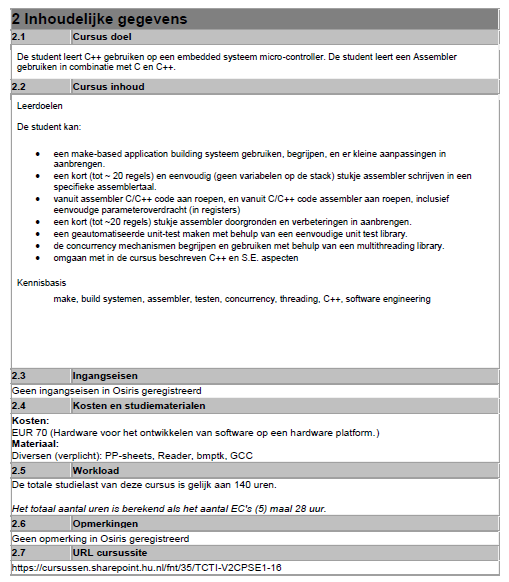
De practicum opgaven staan op sharepoint.

Op sharepoint staan links naar de youtube opnamen van de lessen. De kwaliteit is verre van optimaal maar het is wel verstaanbaar.

# Gegevens studiegids

De gegevens opgenomen in de studiegids (en gepubliceerd via OSIRIS) zijn bindend. Voor deze cursus geldt de volgende OSIRIS-beschrijving.







**Disclaimer**  
“Wij spannen ons in om nauwkeurige en actuele informatie in dit document op te nemen, maar kunnen geen garantie geven dat de beschikbare informatie volledig of juist is. Dit document dient ter algemene informatievoorziening en kan, zonder voorafgaande toestemming of aankondiging, gewijzigd worden. De gebruiker kan aan de informatie in dit document geen rechten ontlenen. Dit document vormt geen onderdeel van de OER en/of de studiegids, en is tevens geen onderdeel van de interne rangorde zoals verwoord in artikel 2 van de OER”