Fontys Hogeschool ICT

Courses overview FHICT English stream

This is the complete overview of all courses offered by FHICT English stream. It contains complete information about all courses from all programs of the English curricula of FHICT for cohorts that started before september 2019. Courses that are used in different curricula are duplicated, but are equal by definition.

Due to the schedules and changes in curricula, not all courses described in this document are actually offered in the period this document is about. In such case, the information may be incomplete or outdated. The correct information is provided in the document covering the period in which a course is given.

Table of contents:

1. Information about Specialisation ICT & Business

1.1. Information about MATH1 Mathematics 1

Content

- sets; algebra of sets
- counting principles; relations; mathematical induction
- logic, propositional calculus, truth tables
- quantifiers: universal, existential, sum, product

Learning outcomes

After completing this module the student is able to:

- calculate algebra of sets
- apply counting principles
- depict relations as sets, and vise versa
- proof a proposition via mathematical induction
- execute propositional calculus
- determine the truth table of a proposition
- calculate the result of these quantifiers: universal, existential, sum, product

Examination and grading

Examination

written exam

Tools allowed

table of laws is attached to the exam papers

Resits and repairs

next semester

Grading

final mark is the mark of the exam

Teaching methods

lectures for theory and practical

Resources

reader + slides

1.2. Information about PCS1 Programming in C# 1

Content

Subjects:

- Console applications
- Primitive data types
- Expressions, assignment statement,
- type conversions, operators
- Input/output
- Selection statements (if, switch and conditional operator)
- Loop statements (while and for loops)
- Classes

Learning outcomes

The student will gather knowledge and competences about programming in an object-oriented way.

Examination and grading

Examination

Exam on paper.

Tools allowed

whatever he/she brings in on paper.

Resits and repairs

exams in the exam-weeks of the odd-numbered blocks

Grading

mark for exam is final mark

Teaching methods

theoretical lessons and practica

Resources

Book Visual C# 2012 How to Program, Paul Deitel and Harvey Deitel

1.3. Information about EDB1 Databases 1

Content

- Introduction into relational databases

- Structured Query Language (select)

- Normalisation

- MS Access

Learning outcomes

The student is familiar with relational databases

The student can apply:

- normalisation

- select-command for answering queries

The student can work with MS Access

Examination and grading

Examination

Written exam.

Tools allowed

None.

Resits and repairs

Next semester.

Grading

Final mark is the mark of the exam.

Teaching methods

Lectures for theory and practical.

Resources

Reader and slides and optionally a book.

1.4. Information about FIS1 Fundamentals for ICT Students 1

Content

In this course we will work on getting your knowledge level to that of a starting IT student by covering the following topics:

- Basic hardware understanding;
- Basic operating system understanding;
- Calculus covering: Binary, hexadecimal en negative numbers;
- Background information about programming;
- Soft skills.

Learning outcomes

- You can explain how the hardware parts of a computer work as a whole;

- You can explain how a CPU works and what it's usage is;
- You can explain how a GPU works and what it's usage is;
- You can explain the different types of memory and their usage;
- You can explain what an OSand what kind of characteristics it can have;
- You can sum up the tasks of an complex OS;
- You can explain how multitasking works in an OS;
- You can explain how an OS handles memory for a processes;
- You can explain why processes cannot use the same resource at the same time;
- You can explain how a processes can communicate with an I/O device;
- You can explain what syntax, semantics and paradigm is;
- You can sum up different types of programming languages;
- You can explain what the usages of different types of programming languages are;
- You can sum up and explain the four main programming paradigms;
- You can describe what a low and high level programming language is;
- You can explain the differences between compiled, interpreted and JIT;
- You can describe the stages of compiled, interpreted, JIT compiled source code to running the application;
- You can explain what the usage advantages of an IDE are;
- You can do conversions between decimal, binary and hexadecimal;
- You can do addition and multiplication in binary;
- You can convert decimal to signed binary.

Examination and grading

Examination

The final grade for FIS1 will be based on the practical assignments and the written exam.

Tools allowed

No additional material is allowed during the written exam (no workbook, no calculator, no mobile phones, no other means are allowed).

Resits and repairs

If the final grade is not sufficient and you want to do a resit you need to redo both the practical assignment as written exam.

Grading

Description	Max. grade	Weight	
Practical assignments (1)		**100**	**30%**
	Group project	*100*	
	Week 5		
Written exam (2)		**100**	**70%**

1: The practical grade only counts if you manage to pass the practical of week 5 with a sufficient. If you did not get a sufficient for week 5 the practical assignments grade will be a 0. Even if you manage to get 100 points for the group project!

2: The written exam will be about the theory given in week 1 up until and including week 6. You can use the slides and your own notes to learn. Keep in mind that the slides have notes which gives an explanation about that slide. The exam will be 90 minutes long.

If the calculated final grade is not sufficient and you need to do a resit you need to redo both the practical assignment as written exam.

NOTE:

- The practical assignments must be completed on time
- You do **not** get any bonus points for the written exam

Teaching methods

In this course you will have a lecture part and a practical part. The lecture will span from week 1 up until and including week 5. As for the practical you're divided in to groups to work on a project. This project will be from week 1 up until and including week 4. As for week 5 and 6 you will work on individual assignments. Week 7 will be the week for recap of the lectures and a practice exam.

Handing in the practical will be done via email to your lecturer.

Course Planning for Students

Week	Lecture	Practical	Deadline deliverables to Tutor
Week 1	Intro module Intro basic hardware CPU, GPU & motherboard	Intro project Form groups and pick subject	Project group and topic
Week 2	Binary & hexadecimal	Exercises for binary and hexadecimal Work on project	Project report and presentation
Week 3	Memory & negative numbers	Project presentation	
Week 4	OS	Project presentation	
Week 5	Programs	Group exercise	
Week 6	Practical	Practical	Group exercise
Week 7	Theory recap	Practice exam	

Resources

- Lecture notes and ppt's
- Internet

1.5. Information about Dutch Introduction Dutch culture and language

Content

This module serves as introduction to living in the Netherlands. What is typically Dutch? In particular, the specific topics that come to discussion in this module are (in arbitrary order):

- Bikes
- Lichtjesroute/glow
- food
- culture shock

- ice skating
- Carnaval
- Orange (Dutch national color)
- Discounts
- Sinterklaas
- Dutch language

Learning outcomes

The student is able to:

- Communicate in Dutch in the following scenarios:
  - Purchasing products in a supermarket.
  - Greeting someone.
- Indentify and describe typical aspects of Dutch culture (See topics).
- Collaborate with other students to produce the final deliverable (video).

Examination and grading

Examination

Dutch finalizes with a presentation of the Video mentioned in the learning goals.

Tools allowed

There is no exam.

Resits and repairs

If a student is more than one lesson absent, or the video of the group is insufficient, the student is required to repeat the module.

Grading

The group will receive a sufficient if he/she satisfies the following requirements:

1 Attend at least six out of seven lessons (participating actively)
2 The spoken Dutch is understandable

The group will receive a good if he/she satisfies the following additional requirements:

1 The spoken Dutch is spoken fluently
2 The video is coherent

Teaching methods

Dutch is taught in a classical approach.

The final deliverable is made in groups.

Resources

Powerpoint presentations.

Language lessons on internet.

1.6. Information about MATH2 Mathematics 2

Content

- Linear Algebra: vector form and equation for lines and planes, matrix calculation, normal vector, dot product, cross product

- Automata: alphabet, words, languages, regular expressions, state diagram; finite state machines, powerset construction

Learning outcomes

After this course, the student is able to

- (Linear Algebra:) calculate the vector form and equation for lines and planes, and apply matrix calculation, normal vector, dot product, cross product
- (Automata:) indicate alphabet, words, languages; determine the language of regular expressions; draw a state diagram of a language; apply the powerset construction of a non-deterministic state diagram

Examination and grading

Examination

written exam

Tools allowed

no additional tools are allowed

Resits and repairs

next semester

Grading

final mark is the grade of the written exam

Teaching methods

lectures with theory and practical

Resources

slides, reader and exercises

1.7. Information about EDB2 Databases 2

Content

- Datamodelling withe the use of Entity Relationship Diagrams

- Relational model

- Data Definition Language

- Data Manipulation Language (insert, update, delete)

- Constraints

- Oracle SQL Developer Data Modeler

Learning outcomes

The student can apply datamodelling with the use of ERD.

The student can transform a datamodel to a relational model.

The student can implement a relational model with the use of DDL, including simple constraints.

The student can modify the content of a relational database with the use of DML.

The student can work with Oracle SQL Developer Data Modeler (or another comparable tool).

Examination and grading

Examination

Written exam.

Tools allowed

None.

Resits and repairs

Next semester.

Grading

Final mark is the mark of the exam.

Teaching methods

Lectures for theory and practical.

Resources

Reader and slides and optionally a book.

1.8. Information about PM Project Management

Content

-

week	lecture	Practical	reader	Deliverables to Tutor	Deliverables to group B
week 1	"Project Statement"	○ make questions for the interview with the formal client for the given case ○ prepare the interview with the formal client	- study chapters 1 and 2	Questions for the client	
week 2	"Project Phasing"	○ project progress meeting tutor (per group, interview with client) ○ make mindmap and project plan version 1	- study chapter 3	Project plan version 1 Mindmap	
week 3		○ project progress meeting tutor (per group, feedback on project plan) ○ make additional questions for the interview with the formal client for the given case		Questions for the client	
week 4	"Project management"	○ project progress meeting tutor (per group, interview with client) ○ make project plan version 2		Project plan version 2	Project plan version 2

week 5		○ make feedback document for received projectplan		Feedback document	Feedback document
week 6		○ project progress meeting with other group (per group, receive your feedback) ○ make project plan Final version		Project plan Final	
week 7					

Learning outcomes

- Understand the importance of project management
- Understand the different tools for managing a project
- Able to ask the right questions to gather the requirements
- Able to reproduce a PID in coming projects
- Understand the different roles in a project

The goal of this course is to learn to write a Project Plan. You need to be able to write a project plan because you will make project plans (1) during your study and (2) at your future career. For example, in the next semester you will have to make a project plan at the beginning of the Project P-Phase (ProP). In this course the focus will be on writing a project plan within your career as a student of Fontys.

Examination and grading

Examination

In order to successfully pass the course, you must fulfil all requirements listed below:

1
Be present at lectures and progress meetings in all weeks (this course lasts 6 weeks). If you cannot attend the lecture/meeting for valid reasons (e.g., sickness), you must notify the teacher via e-mail before the lecture/meeting.

2
All deliverables must be submitted before the deadline, which will be given by your teacher during the lectures.

3
Mini exam in week 8/9 should be passed sufficient >=5.5

4
The Feedback document is marked Sufficient or more by the tutor.

5
Both the exam and the project plan should be graded with at least 5.5 or higher to pass Project Management course.

If you satisfy all assessment requirements, then your end mark for the Project Management course will is derived from the points for the Project Plan. These points are distributed as follows:

Total points = 10 points
1. Project Statement = 5 points
a. Client = 0,1

b. Leader = 0,1
c. Current situation = 1
d. Problem description = 1
e. Project goal = 0,5
f. Deliverables and Non-deliverables = 1
g. Constraints = 0,3
h. Risks = 1
2. Project Phasing = 5 points
a. Visual overview of the phases, activities, timeline, critical path = 1,5
b. Clear described tasks per activities = 2
c. Time estimation per activity = 0,5
d. Milestones per phase matching the activities= 1
-2,0 points for bad layout (title page, font, headings, alignment, etc.)
-1,0 points for a missing Title Page

Final mark for the Project Management course is derived from the points for the Project Plan and the exam.

project plan points	PM mark
9.5 – 10	10
8.5 – 9.4	9
7.5 – 8.4	8
6.5 – 7.4	7
5.5 – 6.4	6
4.5 – 5.4	5
3.5 – 4.4	4
2.5 – 3.4	3
1.5 – 2.4	2
0.0 – 1.4	1

Tools allowed

Books and materials

Resits and repairs

In case Project plan is >=5.5 and exam <5, retake of the exam next semester.

In case Project plan <= 5.5 whole course must be retaken

In case more than 1 class absent, whole course must be retaken

Grading

Total points = 10 points
1. Project Statement = 5 points
a. Client = 0,1
b. Leader = 0,1
c. Current situation = 1
d. Problem description = 1
e. Project goal = 0,5
f. Deliverables and Non-deliverables = 1
g. Constraints = 0,3
h. Risks = 1
2. Project Phasing = 5 points
a. Visual overview of the phases, activities, timeline, critical path = 1,5
b. Clear described tasks per activities = 2
c. Time estimation per activity = 0,5
d. Milestones per phase matching the activities= 1
-2,0 points for bad layout (title page, font, headings, alignment, etc.)
-1,0 points for a missing Title Page

Teaching methods

7 classes of 2 hours

Homework every week

Resources

- Reader: Project Management, an introduction
- Material on Sharepoint

1.9. Information about PCS2 Programming in C# 2

Content

Subjects:

- Windows applications
- Enumeration
- Overloading
- Arrays
- Collections, lists

Learning outcomes

- Windows applications

- Enumeration
- Overloading
- Arrays
- Collections, lists

Examination and grading

Examination

Laptop exam.

Tools allowed

everything on paper and on their own laptop

Resits and repairs

exams in the exam-weeks of the even-numbered blocks

Grading

mark for exam is final mark

Teaching methods

theoretical lessons and practica

Resources

Book Visual C# 2012 How to Program, Paul Deitel and Harvey Deitel

1.10. Information about FIS2 Fundamentals for ICT Students 2

Content

In this course we will continue to work on getting your knowledge level to that of a starting IT student by covering the following topics:

- Testing and debugging;
- DLL's;
- Version control systems.

Besides these topics we will also give an introduction to the streams ICT & Business, ICT & Technology and ICT & Software. By giving background information and practical exercises you should get a feeling of what these streams are about.

Learning outcomes

- Explain the difference between ICT & Business, ICT & Technology & ICT & Software;
- Explain how to gather requirement from a client;
- Visualize a business process in a diagram;
- Create a simple algorithm for a delivery robot;
- What .NET Framework is and how C# fits into it;
- Explain what a Visual Studio project is and what files it has;
- Make use of a DLL in its C# projects;
- Make use of namespaces in a C# project;
- Explain what a version control system is;
- Work with GIT/GIT repository;
- Explain the difference between debugging and testing;
- Explain what a syntax, runtime, logical and efficiency error is;

- Simple debug techniques he/she can apply;
- Apply advanced debugging techniques like break points & watches;
- Create a test plan & report.

Examination and grading

Examination

The final grade for FIS2 will be based on the practical assignments.

NOTE:

- The practical assignments must be handed in before the deadline. Missing a deadline means a grade of 0;
- If plagiarism is discovered you will get a grade of 0 (insufficient) for that deliverable.

Tools allowed

All additional aids are allwed for practical assignments.

Resits and repairs

Re-sit is possible in the next semester.

Grading

You need to have at least a sufficient for deliverables from week 2 & 3 before you get an end grade.

Description	Max. grade	Weight
Week 1 - ICT & Business deliverable	10	30%
Project	10	70%

Teaching methods

This course consists of two hours theory and four hours practical (with two unsupervised) per week. The focus of this course will be about the practical. The theory will be given in week 1 until week 5.

Handing in the practical will be done via email to your lecturer.

Constraints

During week 1 groups will be formed and instructions will be given. If you're not present during this week the teacher will appointed you to a group. If missed week 1 and aren't present in week 2 you're out of the course.

During week 4 you will get a go or no-go about whether you can continue with the course. If you get a no-go you're out of the course.

Besides that you're required to be present for the first class of a week and also the first hour of the second class of the week. You miss more than one theory and one practical class you're out of the course. You're always required to hand in the deliverables before the set deadlines.

Course Planning for Students

Week	Lecture	Practical	Deadline deliverables to Tutor
Week 1	Intro course Intro ICT & Business Requirements gathering Process diagrams	ICT & Business assignment	Before Week 3: a document with the results of the practical

Week 2	Intro ICT & Technology Version control systems - Part A	ICT & Technology assignment	In week 3: show the deliverable to the lecturer
Week 3	Intro ICT & Software DLL	ICT & Software assignment	In week 4: show the deliverable to the lecturer
Week 4	Version control systems - Part B	Start project	
Week 5	Testing & debugging	Work on project	
Week 6	-	Work on project	
Week 7	-	Finish project	Before Week 8: Deliverables for the project
Week 9	-	Project event	

Resources

- Lecture notes and ppt's
- Practical readers & tutorials provided on Sharepoint
- Internet
- Lego Mindstorms

1.11. Information about ICTA ICT Assessment

Content

The ICT assessment is intended to test integrally whether the learning outcomes of the first semester have been achieved. The assessment only takes place if there is any doubt about the learning outcomes or not. -

Learning outcomes

The learning objectives are the same as the learning objectives as described in the other modules of semester 1

Examination and grading

Examination

Type 5 and 6 testing. Assignments (good / suff / insuff) and oral exam (1..10)

Tools allowed

no extra tools

Resits and repairs

next semester

Grading

Assignments and oral exam are graded by two assessors.

Teaching methods

No special education form concerns only the assessment

Resources

all the lesson material of the other semester 1 modules

1.12. Information about EXC Excel

Content

From Excel these functions are learned:

- (Named) Ranges & Tables
- Sorting & Filtering & Totaling
- Simple calculations in ranges and tables
- Relative & absolute cell references
- Charting & chart feeders
- Pivot tables & pivot charts
- Conditional formatting

Learning outcomes

basic handling of Excel as general business tool

· introduction to Excel as data analysis tool to

a. analyze data

b. visualize data

Examination and grading

Examination

aptop exam: solve an Excel case

– to be able to finish in time

- you need to be fluent and experienced
- you need to have good knowledge of all the exercises we did

– no internet access

Tools allowed

None

Resits and repairs

Next semester, during normal Excel

Grading

Mark between 1 and 10.

Teaching methods

7 weeks classes

Resources

Slaying Excel Dragons: book and accompanyingYouTube channel

- Exercises (with solutions) and mockup exams onSharePoint

1.13. Information about PCSB Programming C# for Business

Content

How to use a list

Exception handling

How to use a RFID-reader in combination with a C#-project

How to make a connection to a MySql-database, so you can retrieve information from the database and you can change (for instance: insert, update) information in the database Inheritance Using events

Inheritance

Using events

Learning outcomes

The student will gather knowledge and competences about programming in an object-oriented way.

Examination and grading

Examination

Assignments

written exam

Tools allowed

None

Resits and repairs

exams in the exam-weeks of the odd-numbered blocks

Grading

mark for exam is final mark

Teaching methods

theoretical lessons and practica

Resources

Electronic references

1.14. Information about BEC1 Business Economics 1

Content

Topics that are discussed during the module

- Introduction to financial reports
- Introduction to financial ratio analysis
- Setup a bookkeeping system and calculate profit and equity. Further calculate costs, expenditures and incomes.
- Financial accounting scenario: ledger, journals, day book, subsidiaries, computing
- Special entries: accrual vs cash system, error bookings, interest bookings, private accounts, funds in transfer

- Financial Accounting.
  - Balance Sheet
  - Income statement
  - Retained earinings
  - Basic GAAP rules
- Fixed assets vs. current assets
- Drawings account
- Cash transfers
- Accrued cost
- Annual Closure (Trial Balance)
- Balanced score card

Learning outcomes

-

○ Understand the importance of financial reports
○ understand the different types of financial ratios
○ Able to understand the numbers of the financial ratios
○ Able to compare companies of different

	size based on ratio analysis
	○ Able to setup a basic bookkeeping system
	○ Able to understand Trial balances, balances, profit & loss statements
	○ understand taxation systems

Examination and grading

Examination

- Group assignments which have been assessed by individual contribution and individual portfolios.(40%) and small test in week 5
- Written exam (60%) in week 8

Tools allowed

pen and calculator

Resits and repairs

Next semester retake

Grading

- Individual feedback on assignments
- Group feedback on assignments and presentations
- Individual feedback on small test in week 5
- Weeks of presentations are mandatory

Teaching methods

7 classes of 4 hour with theory, presentations, assignments

Resources

-

From bookboon

- The accounting Cycle (PDF file)
- International financial accounting. (PDF file)
- Provided slides

1.15. Information about WEB1 Web Development 1

Content

In this module we will study the basic technologies that allow us to build a web application; you will study front-end as well as back-end technologies. During this course students will build a web application from scratch applying a logical process.

Learning outcomes

The student will be able to:

- Understand the main technologies and tools involved in Web development.
- Provide insight in the design and development process of a web application.

- Get to know the following front-end technologies: HTML, CSS and JS.
- Get to know the following back-end technologies: PHP, MySql

Examination and grading

Examination

The grade of *The Project* is the final grade of WEB1. For each deliverable (A, B, …, F) you will get a grade between 1 and 10. Calculating the final grade for *The Project* occurs as described in the table below and the criteria underneath it.

There will be *Go* and *No-Go* sessions. This is determined by your teacher at any point in time during the course. A *No-Go* means your group cannot proceed any further and must leave the course receiving no final grade as consequence. If your group misses a deadline you get 0 points for that deliverable and your final grade will be decreased by 1 point. If your group misses more than one deadline you automatically get a *No-Go*.

Description	Related Deliverable(s)	Weight on final grade
Design Document	A	20%
Front End development	AVG (B,C,D)	50%
Back End Development	AVG (E,F)	30%

Tools allowed

- GIT
- XAMPP, WAMP or AMPPS
- w3Schools.com
- *HTML & CSS design and build websites*, John Ducket

Resits and repairs

If less than 5.5 points is achieved in this module, the student needs to retake entirely the whole module.

Grading

Description	Related Deliverable(s)	Weight on final grade
Design Document	A	20%
Front End development	AVG (B,C,D)	50%
Back End Development	AVG (E,F)	30%

Teaching methods

The approach used in this course is *Learning-by-doing*. Therefore, this course is compound of 1 theory hour and 3 practical hours per week, at school. Besides, the student will have to spend a minimum of 8 hours (self-study) in the matter per week. The theory lectures explain what a certain technology is, the purpose of it and the essentials on how to use it. It is the student task to go deeper and study further in detail such technologies so that she is better prepared to tackle the issues that might appear during the practical assignment.

Furthermore, the student is expected to have the required knowledge previously stated to guarantee a smooth flow of the course and avoid falling behind and or into frustration.

The practical of this course is about building a web application which is called *The Project* (see “The Project” section). Since web development is mainly a collaborative process you will work in groups of 2 people. *The Project* has weekly deliverables (see course planning).

Resources

HTML & CSS design and build websites, John Ducket

This module uses internet resources as well. Such resources have been placed in the end section of this document (resources section).

RESOURCES:

Week 1: PROCESS & DESIGN

Topic	Source
Process & Design of a website	Chapter 18; *HTML & CSS design and build websites*, John Ducket
Installation of tools, setting up environment	Click here

Week 2: HTML

Topic	Source
HTML tutorials, examples, etc.	http://www.w3schools.com/html/default.asp
World Wide Web Consortium (HTML standard)	https://www.w3.org/standards/webdesign/htmlcss

Week 3: CSS

Topic	Source
CSS tutorials, examples, etc.	http://www.w3schools.com/css/default.asp
World Wide Web Consortium (CSS standard)	https://www.w3.org/standards/webdesign/htmlcss

Week 4: Js

Topic	Source
JS theory	https://developer.mozilla.org/en-US/docs/Web/JavaScript
JS tutorials, examples, etc	http://www.w3schools.com/js/

Week 5: PHP

Topic	Source
PHP official site	http://php.net
PHP Arrays	http://w3schools.com/php/php_ref_array.asp
PHP Superglobal variables	http://www.w3schools.com/php/php_superglobals.asp

Week6: PHP

Topic	Source
PHP Cookies	http://php.net/manual/en/features.cookies.php
PHP Session	http://php.net/manual/en/book.session.php
PHP PDO class	http://php.net/manual/en/class.pdo.php
PHP PDOStatement class	http://php.net/manual/en/class.pdostatement.php
PHP PDOException class	http://php.net/manual/en/class.pdoexception.php
PHP File uploading	http://php.net/manual/en/features.file-upload.php
PHP File system	http://php.net/manual/en/book.filesystem.php

1.16. Information about PO/PD1 Professional Orientation and Personal Development 1

Content

The topics for this module are:

1 company visits,
2 cultural awareness,
3 presentation skills
4 introduction research framework

Learning outcomes

This course has two goals: To orient on the actual ICT work field and to improve the communication skills.

Examination and grading

Examination

In order to successfully pass the course, you must fulfil all requirements listed below:

1 Be present at lectures and progress meetings in all weeks . If you cannot attend the lecture/meeting for valid reasons (e.g., sickness), you must notify the teacher via e-mail before the lecture/meeting. An additional assignment will be given in week 8.
2 All deliverables must be submitted before the deadline, which will be given by your teacher during the lectures.

Tools allowed

no end exam.

Resits and repairs

if an assignment is insufficient the student can improve this once during the course.

if a student did miss a meeting, an additional assignment will be given in week 8.

Grading

If the student satisfy all assessment requirements, then your end mark for the PO/PD course will be “Voldoende”, meaning Sufficient.

Teaching methods

POPD1 works with group assignments and individual assignments.

Resources

powerpoint presentations

websites

DOT research documentation

1.17. Information about STAT

Content

Basic knowledge about statistical expressions within data mining and big data. Needed to select the relevant function for a particular analysis and for the right interpretation of the results coming out of the analyzed model.

Learning outcomes

The student:

- Can understand and apply Normal Distribution;
- Is able to test hypotheses (zero-hypotheses and alternative hypotheses);
- Can calculate an Anova (Analysis of variance, one/multi-dimensional)
- Can process a correlation and regression analysis;
- Can apply Chi-square on a series of observations;
- Can apply index numbers on time series;
- Can create a universal Excel help sheet (which can assist solving statistical questions)

Examination and grading

Examination

Exam and Excel help sheet (both individual)

Exam will take place in week 9. Individual Exam (70%), grade <0-10> Excel help sheet (30%), grade <0-10> Both parts must be sufficient (grade >= 5,5) .

Tools allowed

MS Excel (and helpsheet).

Resits and repairs

Resit of the complete course is done via an individual exam in week 16, if applicable the grade for the Excel help sheet will not be taken into account anymore.

Grading

Graded between 1 and 10.

Teaching methods

Lectures and Practical Guidance

Resources

Business Statistics for Non- Mathematicians, Sonia Ann Taylor (obligated)

MS EXCEL (incl. data analysis toolpack), Fontys application portal

1.18. Information about BEC2 Business Economics 2

Content

- Introduction to financial reports
- Introduction to financial ratio analysis
- Setup a bookkeeping system and calculate profit and equity. Further calculate costs, expenditures and incomes.
- Financial accounting scenario: ledger, journals, day book, subsidiaries, computing
- Special entries: accrual vs cash system, error bookings, interest bookings, private accounts, funds in transfer

Learning outcomes

- Understand the importance of financial reports
- understand the different types of financial ratios
- Able to understand the numbers of the financial ratios
- Able to compare companies of different size based on ratio analysis
- Able to setup a basic bookkeeping system
- Able to understand Trial balances, balances, profit & loss statements
- understand taxation systems

Examination and grading

Examination

- Group assignments which have been assessed by individual contribution and individual portfolios.(40%) and small test in week 5
- Written exam (60%) in week 8

Tools allowed

pen and calculator

Resits and repairs

Next semester redo of the exam or retake of the course in case practical failed

Grading

Presentations / Assignments (40%)

Exam written week 8 (60%)

Teaching methods

7 Classes of 4 hours

Theory, assignments, feedback, presentations

Resources

- The Accounting Cycle; Authors Larry M. Walthr; Christopher J. Skousen; ISBN 978-87-7681-486-1
- Slides and practical assignments on the SharePoint.
- Internet sources
- The essence of financial accounting - Leslie Chadwick - ISBN: 0-13-356510-6

1.19. Information about PO/PD2 Professional Orientation and Personal Development 2

Content

The topics for this module are:

1 information skills,
2 writing skills
3 DOT research framework: triangulatie

Learning outcomes

This course has three goals:

Improvement of a research project by learning:

- how to find information and when is it valid and reliable

- how to improve writing skills

- how the use of triangulation during a research project

Examination and grading

Examination

In order to successfully pass the course, you must fulfil all requirements listed below:

1 Be present at lectures and progress meetings in all weeks . If you cannot attend the lecture/meeting for valid reasons (e.g., sickness), you must notify the teacher via e-mail before the lecture/meeting. The assignment has to be done anyway.
2 All deliverables must be submitted before the deadline, which will be given by your teacher during the lectures.
3 No plagiarm allowed. All personal written essays will be checkt by Euphorus: copy paste will be considered as fraude.

Tools allowed

no end exam.

Resits and repairs

If an assignment is insufficient, the student can improve this once during the course.

Grading

If the student satisfies all assessment requirements, then the end mark for the POPD2 course will be “Voldoende”, meaning Sufficient.

Teaching methods

POPD2 works with a group assignment for the research topic and individual assignments for the rest.

Resources

powerpoint presentations

websites

DOT research documentation

1.20. Information about PROP Project P-phase

Content

Project P-phase (ProP) is a second semester module in which the students work in a group. The group is tasked with the development of a software solution that will facilitate the management of an event.

Pre-requisites:

- Completed the modules with a sufficient: EDB1, EDB2, PCS1 and PCS2.

Learning outcomes

The student should be able to:

- Design a software solution and plan a project based on a case.
- Implement the final product according to the design and plan.
- Implement the applications, website and database as one coherent system.
- Justify decisions that are made in the project.
- Apply version control in GitLab to the project according to FIS2 guidelines.
- Communicate effectively with the client.
- Conduct meetings in orderly/organized fashion.
- Collaborate effectively with group members within a project.

Examination and grading

Examination

The process is assessed by the tutor and a client (second tutor). At the end of the project, the group presents their solution to the client and reflects on the project. After this presentation the tutor and client grade the group. The tutor and client could decide to apply individual grades to (part of) the group.

Tools allowed

Not applicable.

Resits and repairs

If a student failed ProP, he/she will have to restart it during the next semester. The student will have to participate in a new group.

Grading

The final grade ranges between 1.0 - 10.0.

Criteria can be found in the current version of the student workbook (appendix C: Assessment Sheet).

The grade is dependent on the quality of the process and deliverables of the project.

Teaching methods

The student will work in group formation on the project. Teachers tutor this process.

Resources

Students can find a workbook in the shape of a PDF file on sharepoint.

1.21. Information about NETWB

Content

- Network Layered Model, TCP/IP
- IP Protocol, IP Addressing, IP Subnets, ICMP, ARP
- IP Routing
- TCP/UDP protocols
- DHCP/NAT protocols
- DNS protocol
- Network configuration in Linux environment

All above mentioned protocols will be practiced in assignments

Learning outcomes

- The student can identify different layers of the TCP/IP protocols.
- The student can design network drawing including IP addressing
- The student can use basic Linux networking commands to configure a small IP network
- The student can explain different sort of routes used in static IP routing
- The student can interconnect different nodes of a small network by analysing and using different types of static routing
- The student can explain difference between static and dynamic IP addressing
- The student can explain the use and the phases of DHCP protocol
- The student can demonstrate the working of DHCP protocol in a networking scenario
- The student can explain the use of NAT protocol
- The student can demonstrate the working of NAT protocol in a networking scenario
- The student can configure basic DNS server in a Linux environment
- The student can name the basic types of DNS records
- The student can explain the basic working of the DNS protocol
- The student can name the basic advantages of the IPv6

Examination and grading

Examination

- all assignments have to be completed and graded in Canvas
- written exam

Tools allowed

None

Resits and repairs

Every semester retake is possible

Grading

- all assignments have to be completed and graded in Canvas
- written exam

Teaching methods

7 classes

Resources

Olivier Bonaventure: Computer Networking : Principles, Protocols and Practice Netkit Tool, Wireshark, VMWare

1.22. Information about SDB1 System Development 1

Content

- Recall the **basic types of computer-based systems** that a systems analyst needs to address.
- Realize what the many **roles** of the systems analyst are.
- Comprehend the fundamentals of three development methodologies: **SDLC**, the **agile** approach, and **object-oriented** systems analysis and design.
-

Understand and be able to apply concepts of information systems, i.e.

- An overview of information systems.
- An introduction to information technology.
- The concept of “application.”
- Information systems as products.
- The business of developing information system products.
- Information system as the infrastructure of the business.
- The enterprise of software development.

Understand and be able to apply the basic concepts of object orientation, i.e.

-

Real-world objects.

-

Object identity.

-

Object’s attributes and operations.

-

Classes and classification.

-

Encapsulation and information hiding

-

Object interface.

- Aggregate and composite objects.
- inheritance and polymorphism.
- Object-oriented technology.
- Object-oriented modeling and the Unified Modeling Language (UML).
- Understand and be able to apply concepts of methodology, i.e.

- Fundamental concepts and building blocks of methodology.
- Benefits and risks of methodology.
- Software development methodologies.
- Modeling concepts and software development.
- Project management concepts and tools.

Learning outcomes

- Understand the different system development methodologies
- Know when to use the different type of methodology based on criteria
- Student is able to apply theory in cases
- Student is able to describe the different methodologies
- Student can work in an agile project
- Student understand basic models of PM

Examination and grading

Examination

- Groups: Assignments (Homework) and Presentations (WF: 0.4)
- Tests (WF: 0,6)

Tools allowed

None

Resits and repairs

Retake every semester. If passed for practical only exam retake

Grading

- You are actively present in all classes, which means:
  - you arrive on time in class
  - during assignments and presentations, you are active on the topic at hand,
  - during theory you have an active attitude,
  - you have prepared well before class and do your homework.
- One class missed with valid reason is additional assignment.
- Two classes or more missed. No grade for practical.

Teaching methods

6 classes of 4 hours theory and practical assignments / presentations

1 Guest lecture Agile / Scrum

Resources

Book: Ashrafi

- **Object Oriented Systems Analysis and Design**
- Noushin Ashrafi and Hessam Ashrafi
- First edition (New International Edition)
- Chapters: 1, 2, 3

Book: Kendall

- **Systems Analysis and Design**
- Kenneth Kendall and Julie Kendall
- Ninth edition (Global Edition)

Chapters: 1, 2

1.23. Information about EDB3 Databases 3

Content

Subjects:

- Principles of relational databases
- Design and implementation of complex queries in SQL
- Data manipulation.
- ACID
- Concurrency control
- Oracle SQL practical

Learning outcomes

The student understands and is able to create complex SQL queries (SELECT statement) using

-Functions

-Joins

-Subqueries

-Views.

The students understands aspects of Managing Multiuser Databases.

Examination and grading

Examination

Written exam + practical

Tools allowed

None.

Resits and repairs

Next semester.

Grading

Final mark is the mark of the exam.

Teaching methods

Lectures for theory and practical.

Resources

Reader and slides.

Database Processing, David Kroenke & David Auer, 12th edition

1.24. Information about SDB2 System Development 2

Content

Understand and be able to apply the concepts of requirements gathering, i.e.

- Define requirements
- Requirements discovery
- Classifying requirements
- Techniques for eliciting requirements
- Managing requirements
- The case history of Walden Hospital, the main source for examples in this book

Understand and be able to apply the concepts of object-oriented analysis in the area of domain analysis, i.e.

- The three components of problem solving.
- The problem space vs. the solution space.
- Requirements vs. product specifications.
- Domains and their boundaries.
- Identifying domain concepts for analysis and modeling.
- Domain dictionaries and domain catalog.
- Identifying and organizing business rules.

Understand and be able to apply the concepts of object-oriented analysis in the area of behavioral modeling, i.e.

- What use case modeling is and is not.
- The four components of a use case.
- The basic elements of use case diagram.
- How to transform concepts from domain analysis into use cases.
- Identifying prominent actors.
- Identifying major use cases.

- The context diagram.

- Structuring and developing use cases through templates.

- When and how to generalize actors.

-

When and how to extend the functionality of a use case.

- When and how to reuse use cases.
- When and how to generalize use cases.
- The features and the purpose of use case diagram.
- When and how to join or divide use case.
- Using activity diagram to clarify the logical flow of use cases.
- Use case modeling as a framework for development activities.
- Managing details by creating supplements to use cases.

Learning outcomes

- Student is able to choose between data gathering methods
- Student understands the problem space and solution space
- Student is able to model use cases
- Student has knowledge of all related terminology
- Student is able to gather requirements and model them for business
- Student is able to translate information gathering into models readable for business people

Examination and grading

Examination

- Written exam (60%)
- Theoretical and practical assignments and presentations (40%)

Tools allowed

None

Resits and repairs

Retake every semester or resit of exam every semester

Grading

Review of the exam

Presentations in class

group assignments

presence during guest lecture

Teaching methods

7 classes of 4 hours

Resources

- **Book:** Object Oriented Systems Analysis and Design, Ashrafi & Ashrafi 1e 2009 ISBN: 978-1-29203-960-2

- chapters 4, 5, 6, 7 Ashrafi

1.25. Information about EBUS E-Business

Content

- *E-tech: the backgrounds about the function and structure of networks and servers in general and specially of the internet.*
- *E-com: the economical, organizational and commercial consequences of doing business over the internet (e-commerce).*
- *E-prog: basic knowledge about the most used languages and tools for building a website and a web shop.*
- *E-project: a project in which a web shop is being build.*

Learning outcomes

- *Technical knowledge about the function and structure of networks and servers in general and specially of the internet.*
- *Knowledge about the economical, organizational and commercial consequences of doing business over the internet (e-commerce).*
- *Knowledge about the most used languages and tools for building a website and a web shop.*

Examination and grading

Examination

- *E-tech: exam*
- *E-ecom: case study and presentation on an e-commerce related subject.*

E-project: a ready made and functional web shop

Tools allowed

none

Resits and repairs

retake every semester

Grading

Review of exam and case study. 2 Separate marks

Teaching methods

7 classes of 4 hours

Resources

- *lecturer notes and ppt's,.*
- *books :*

1.26. Information about QP Quality Principles

Content

Introduction

Topics that are discussed during the module

-

Introduction Quality Management

- Perspectives of Quality Principles
- Software: CMMI, SPI,
- Production: Kaizen, Lean, Six Sigma
- Financial: IT Auditing, IFRS, BASEL II
- Compliancy: Sas-70, SOX, Laws, Pharma. Regulations
- Controls: Porter, Kaplan, Balanced Scorecard, Deming
- Administration: ISO 9001:2000
- Security: ISO 27001
- Ethics: Codes of conduct, Declaration on oath, anti-bribary acts
- Risk Management

Learning outcomes

- Understand the importance of quality systems
- Student is capable of applying quality models on real cases
- Student has developed a critical attitude regarding quality systems and the achievableness of the implementation
- Student works analytical and has the ability to advice management based on research
- Student has developed skills in writing an essay
- Student has developed a critical view environment and is able to judge impacts of measures and regulations

Examination and grading

Examination

- Groups: Assignments (Homework), Presentations and individual essay (WF: 0.7)
- Written exam (WF: 0,3)

Tools allowed

- Books
- Laptop
- Internet

Resits and repairs

Retake every semester of the written exam

Grading

- **Assessment of Tests, Assignments and Presentations**

At Fontys ICT we want to stimulate that students are very active in their classes.

Group Assignments and Presentations and individual essay are graded with 70% of the final score if:

You are actively present in all classes, which means:

- you arrive on time in class
- during assignments and presentations, you are active on the topic at hand,
- during theory you have an active attitude,
- you have prepared well before class and do your homework.

You're individual essay contains:

- 3 chosen topics applied to the company chosen
- In Euphorus <10% defects and plagiarism
- Has your critical view on the topics
- Contains a strong stakeholder analysis and quadrant
- Contains a worked out Porter Value Chain and balanced scorecard applied to your company with critical underpinning
- Comply with DOT framework research model
- APA References used
- Your point of view as CEO of the company with "out of the box" solutions.

Teaching methods

7 classes of 4 hours of which one class a Guest Lecture about Lean Six Sigma / Kaizen

Resources

Resources on Sharepoint published

1.27. Information about OODB Object Oriented Development for Business

Content

UML Use-Case

UML Class diagrams

UML sequence diagram

Using files

Unit testing

Develop an application with the help of UML

Learning outcomes

The student will gather knowledge and competences about UML and implementing in an object-oriented way.

Examination and grading

Examination

Written exam

Tools allowed

nothing except your own brains

Resits and repairs

exams in the exam-weeks of the odd-numbered blocks

Grading

mark for exam is final mark

Teaching methods

theoretical lessons and practica

Resources

Same book as used in the course OOD1 or book used in the SD courses

1.28. Information about IPHB4 BI-Dutch

Content

To start with, the entry requirement for semester 4 Business is Completion of the foundation year (propedeuse).

In semester 4 the Body of Knowledge (BOK) consist of Business Administration, Information Technology and Personal Professional Development. The Professional Task for Business is a meaningful and authentic educational form that fosters integration.

Fontys IT's vision is to motivate students by allowing them to be entrepreneurial, to enable them to discover their talents, to deploy, to further develop and eventually grow to the professional who they can and want to be.

To this end, there is now the opportunity for all students to:

- make choices with regard to the perspective of the professional task, timing of assignments, the role within the professional task, etc.

- to participate in the Business Academy in all spring semesters.

The Professional Task for Business shows the whole supply chain in which all BOK subjects are fully integrated. In 18 weeks you and your team set up a company. Here you learn to deal with DSDM / Atern, you will work on the business case, run a selection process, define a simple system architecture, determine on the basis of the business blueprint the requirements and determine the economic feasibility. The business processes are customized in Exact, so that procurement, production, storage and end product sales can take place. This takes into account Organizational Change Management aspects together with prototyping. You also learn to set up and interpret Key Performance Indicators (KPI's), for adequate management reports. These will be evaluated and a written reflection making use of the rubrics will be made. Within the given context, you have the freedom to make choices and decisions in terms of the structure and design of the organization, how the Exact system will be implemented, the way people react to the market movements etc. Many topics that were covered in the various disciplines come back in practical form in this professional task, such as: DSDM, Exact, Requirements, BOM, MRP, Testing, KPIs and Statistics, Reporting, Research.

The Business Academy (takes place only in the second half of the educational year) is for entrepreneurial talent oriented students, an alternative assignment instead of the professional task. This concerns in the period January - July of each year, organizing the annual Fontys FHICT Business Academy by students from different academic years. Students must apply for this. The final group itself will have to come up with a plan and have to choose a theme. For this, a virtual company will be set up in which all business functions such as marketing, finance and logistics are involved. This will be accompanied by teachers.

Where the student in semester 2 focusses on company processes within a small size company and semester 3 will let the student get acquainted with logistics processes within a medium size company, semester 4 will deal with large companies and their supply chains. All to prepare yourself for your internship in semester 5 where you are mostly operating independently, being on your own.

The content of the different subjects within this semester 4 course in more detail:

Supply Chain Management: Project Planning, Production Planning, Master Scheduling, MRP, Capacity Management, PAC, Purchasing, Forecasting, Inventory Management, Order Quantities, Warehouse Management, Physical Distribution, Products and Processes, Lean Production, TQM, Supply Chain Management, Supply Chain Strategies.

Governance, Risk and Compliance: Principals and instruments.

Administrative Organization: Function and basic concepts of administrative organization and internal control. Principles and basic techniques of the AO and internal control. Impact of AO on the process design and process control. The administrative organization of the function areas, Purchasing, Inventory, Sales and Production. Aspects of the external audit.

Annual Report Analyses: Financial statements (balance sheet and profit & loss), calculating ratios (e.g. turnover, liquidity, solvency, profitability), interpreting ratios.

ERP: Exact, process analysis, transaction processing of purchase and sales orders, invoices (incoming and outgoing), business processes.

Power BI & Applied Statistics: Dimensional modelling, extracting data from various data sourc es, create performance indicators (in DAX and PowerPivot), verifying results, explaining professional products to peers and domain experts.

Personal Professional Development: Cultural awareness, entrepreneurial attitude & behaviour, project management, internship preparation, research.

Learning outcomes

The student:

Professional development

Analysis

1 is able to formulate appropriate research questions and secondary questions.
2 is able to implement the appropriate research strategies and techniques in the correct manner.
3 is able to substantiate his own work with regard to method and content with the research cycle.
4 is able to set up and carry out a research methodology, with the chosen methods forming a research pattern.

Note

Student is able to collect and interpret relevant data (primarily in the field) with the aim of assessing and forming an opinion that is partly based on the weighing of relevant social, international, scientific and ethical aspects and uses this to demonstrate a scientific research attitude.

Communication

1 is able to report on a practical assignment both verbally and in writing, expressing himself in a cogent, structured text with clear argumentation of the various positions.
2 is able to work effectively alongside other student peers, teachers and professionals in a company or institution.

Note

Student is able to communicate information, ideas, and solutions to an audience consisting of specialists or non-specialists, is able to collaborate and demonstrates (inter-) cultural awareness.

Learning skills

1 is able to describe his professional talents and development ambitions in relation to the field of ICT.
2 is able to reflect and receive feedback based on his own performance in the field of ICT.
3 takes initiative, has an independent attitude and is able to work independently and in a result-oriented manner on an assigned task outside of the university.

Note

Student possesses the learning skills required in order to undertake a subsequent study that assumes a high degree of autonomy, and is enterprising.

Supply Chain Management (SCM)

1. is able to recognize and describe the concept of Supply Chain.

2. is able to recognize, evaluate and shape Supply Chain strategies.

3. is able to apply a supply chain strategy to a (fictional) company.

Governance, Risk and Compliance (GRC)

1. is able to recognize and describe the corporate governance of a company.

2. is able to apply risk assessment for a (part of a) company.

Administrative Organization (AO)

1. is able to apply certain AO tools in a (fictional) company and to set up an AO manual.

Annual Report Analyses (ARA)

1. is able to analyze an annual report and calculate financial ratios.

2. is able to explain the meaning of the outcomes of the ratio’s and can describe the deasoning of the different outcomes of the ratio's.

ERP (ERP/Exact)

1. is able to describe the concept of ERP and can describe how to implement such a system.

2. is able to configure the main SCM processes for a (fictional) company in Exact.

3. is able to describe the financial consequences of the transactions in an ERP system.

Power BI & Applied Statistics (Power BI & AST))

1. is able to set-up a dimensional (reporting) data model from various data sources.

2. can create DAX measures in Power BI. The student specifically demonstrates understanding of the CALCULATE () function.

3. is able to verify measures and dashboard results using SQL, R/tidyverse/Shiny and (traditional) Excel.

4. can conduct a simple time series analysis applying smoothing and decomposition into trend, (seasonal) pattern and noise.

5. can explain his or her professional products to peers and domain experts.

Professional Task (PTB)

1. can bring all competences acquired during semester 4 into practice.

2. can consistently apply the Plan, Do, Check, Act cycle in order to show growth.

"As a completion of the professional task and as a supplement to your PO substantiation in your portfolio, you pitch your realized solution as a group during our own event: Business Showcase Professional Tasks. Inspire each other, teachers and Partners in Education. The event takes place in lesson week 17 on Wednesday afternoon June 17 at location R1_OIL. "

Examination and grading

Examination

During the semester you work in various ways on varying themes and topics. You show the teachers how you learn, develop yourself and use feedback, feedup and feedforward. You regularly validate* your proces of learning and development with the theme teachers and semester coach. The acquired knowledge and experiences are applied into your professional products. By using the feedback, -up and -forward you create professional products to prove that the learning objectives of this semester have been achieved. In the end, you regularly have your professional products validated* by the teachers and semester coach as well.

Both the validated outcomes of your learning and development process, and the validated professional products, are collected in your semester portfolio. At the end of the semester an assessment takes place in the form of a portfolio check in which the integrared semester assessment is taken.

The portfolio check will only be executed when the portfolio was delived completely and before the deadline. No portfolio check results automatically in unsatisfactory. The portfolio check is an inspection type A.

*Validation of your proof can be achieved by:

- formative feedback from the theme teacher, in writing,
- formative feedback from the theme teacher, orally and then documented by the student in Canvas - initiated by the student - validated by the teacher,
- formative feedback in the form of an U-S-G-O (Unsatisfactory, Satisfactory, Good, Outstanding).

Tools allowed

NA

Resits and repairs

Our educational system is giving you the opportunity to prove reached learninggoals during the semester. You receive multiple times (longitudinal) feedback while repetitively showing your results to the teachers, so they have a good view on your product and progress. You are expected to be present regularly (≥ 80%) and to ask feedback frequently (≥ per two weeks). You apply the feedback on your work and have the teacher validate your product. When insufficient presence, asking and applying the feedback and validating the feedback, this can not be corrected the last week(s), as a profound view of your learningproces would be missing.

The portfolio check is part of the Practice-related Testing category (see Article 28 of the OER). The portfolio check can **not** be retaken within the semester. Retake and/or repair is only possible in the next half year, by means of restart or customization.

Grading

The summative assessment is expressed, at the end of the portfolio check, in U-S-G-O (Unsatisfactory, Satisfactory, Good, Outstanding). An insufficient result leads to restart or a customized semester.

Teaching methods

Professional task

Briefing and guided practicum to also stimulate a professional and entrepreneurial attitude and behaviour. The student sets up a company and needs to collect and process essential information, make choices, make decisions and answer for these.

Subjects

Self-study, lectures and seminars which are elaborated in groups on assignments and case studies to also stimulate an entrepreneurial attitude and behaviour. In between quizzes that are intended only to assist the student in the ingestion of the theory (knowledge and understanding). This theoretical basis is needed to apply them in assignments and case studies.

Resources

Books (Status: x = mandatory, o = to be purchased after consultation)

Theme	ISBN	Title	Press	Author	Publisher	Price	Status
Supply Chain Management	978-01-3233-761-8	Introduction to Materials Management	2008	Tony Arnold e.a.	Pearson New International Edition	Free	X
Business Economics	978-87-7681-984-2	Risk Management Made Easy	1st	Andy Osborne	Bookboon.com	Free	X
Business Economics	978-87-7681-737-4	Corporate Governance and International Business	1st	David Crowther and Shahia Seifi	Bookboon.com	Free	X
Business Economics	978-87-403-0956-0	Construction Financial Management	3de	S.L. Tang	Bookboon.com	Free	X
Statistics	978-	Statistics in	1st	A. Buijs	Noordhoff	± € 33	0

	90-01-86121-6	20 steps			Uitgevers		

Software

Business Informatics	Excel	Microsoft	Windows	2016	Eng	Incl. PowerPivot, PowerQuery and Power View add-ins

Internet

Dutch / English language	Internet	Hogeschool-taal	± €75	https://www.hogeschooltaal.nl/login/student/

1.29. Information about Internship

Content

During the four year bachelor course Information & Communication Technology you will work as a trainee in a company (profit or non-profit).

This internship comprises 90-100 working days (planned in the second semester of the third year). Take enough time to prepare yourself!

In this brochure you will be informed about the preparation, the realization and the evaluation of the internship project period.

Learning outcomes

- Student is able to apply all teached knowledge during the graduation project
- Student is capable of applying the DOT framework research model
- Student can critical assess the environment of the company and set frequirements and scope by interviewing business people
- Student has the capability to advice the company with recommendations
- Student is capable of working by him / her self
- Student takes an entrepeneural attitude
- Student is able to adapt to the business and codes of conduct and regulations in the company

Student can finish his / her product / project within 90-100 days

Examination and grading

Examination

internship project > 5.5 (40%)

Presentation before university tutor and company tutor > 5.5 (20%)

Process Report > 5.5 (40%)

Tools allowed

Not applicable

Resits and repairs

Student can, after approval of the FHICT Examboard, repair either the presentation or the report but not the work. All parts must be 5.5 or higher. In case of repair a maximum mark of 6 is awarded.

Grading

Student is assessed at the company by the university tutor and the company tutor for the work contents and attitude at the company

Student is assessed and questioned during his / her presentation before the university tutor and the company tutor

Student is assessed on his / her process report

Teaching methods

University tutor visits the company and the student at least two times. One time between week 4-6 and one time to rehearse the final presentation and discuss the marks

Resources

all learned at Fontys ICT

1.30. Information about 29ita-18

Content

It Auditing will introduce the basic principles of IT auditing to the student. Models and regulations will be discussed and through several assignments and cases the student will learn the broadness of the field of work of an IT Auditor. The basics finds its inheritance in the financial processes. During the course the student will understand the importance of understanding the business processes and is able to define and model processes in tooling like Bonitasoft.

Learning outcomes

Learning outcomes. The student will be able to:

- Understand the impact of an IT- Audit
- Understand and apply the business ethics
- Capable of writing an advice report on high level topics
- Understand the components of an IT-Audit and understand the relation to business processes and financial processes
- Perform an IT-Audit on a financial or outsourcing process
- Understand the principles of Compliancy and due diligence
- Understand and apply internal controls
- Understand the different types of audits
- Understand the relation to cyber security
- Understand the different models of outsourcing and how to audit outsourcing
- Understand the principles of CISA
- Perform a risk assessment on a practical case and explain the consequences
- Understand and apply IT governance
- Understanding IT strategies
- Is capable of advicing a company on the best strategy towards IT Auditing

Examination and grading

Examination

During classes students deliver several homework cases and present them. Further an end essay must be written and a presentation given in week 7 of the course.

A written exam is optional

Tools allowed

During class all given materials, Sharepoint and a Lynda.com videotraining

At the written exam nothing.

Resits and repairs

Every semester the course is given a possibility to repair is possible.

Grading

Review of the quality of the handed in homework, essay and presentations

Teaching methods

7 x 4 hours classes with lots of discussions. 2 hours can be used for the homework and the final essay.

Resources

Sharepoint and during the class. In a later stadium through Canvas.

1.31. Information about 29ct-18

Content

The objective analysis of facts from a judgement. One of the most important skills for students to develop and to fade out Bias and find root-causes for problem definitions. The course will be focusing on learning new skills and techniques to critical assess problems, requirements and to reach the actual goals of customer demands. The basics are fundamentals for journalism and writing a theoretical thesis and scientific research.

Learning outcomes

Learning outcomes The student will be able to:

- Understand the impact of BIAS
- Critical assess problems and perform root cause analysis
- Model root-cause analysis
- Explaining an Issue or Problem. Upon completion of a faculty-designed intervention, students will demonstrate an increased ability to explain an issue or problem comprehensively.
- Employing Evidence/Information Effectively Upon completion of a faculty-designed intervention, students will demonstrate an enhanced ability to employ evidence/information in conducting a comprehensive analysis of an issue or problem.
- Analyzing Contexts. Upon completion of a faculty-designed intervention, students will demonstrate an enhanced ability to analyze contexts when presenting a position on an issue or problem.
- Describing Your and Others Perspectives Upon completion of a faculty-designed intervention, students will demonstrate an enhanced ability to describe he/her perspectives along with those of others associated with the situation.
- Formulating a Thesis/Hypothesis Upon completion of a faculty-designed intervention, students will demonstrate an enhanced ability to formulate a thesis/hypothesis that takes into account the complexity of an issue or problem.
- Drawing Logical Conclusions Upon completion of a faculty-designed intervention, students will demonstrate an enhanced ability to draw logical conclusions and implications from the analysis of an issue or problem.

Examination and grading

Examination

Several cases during the class, discussions, presentations and a final essay on a self chosen topic with critical reflection and the use of the verbs during the course.

In week 7 demonstrating the essay and critical assess the chosen topic.

Tools allowed

Laptop, and give materials. Lynda.com video course

Resits and repairs

Every semester the course is given a possibility to repair is possible.

Grading

Final essay + presentation

Optional written exam based on a case.

Teaching methods

7 classes of 4 hours containing theory and practical cases

Resources

Sharepoint handed out in the class and videocourse Lynda.com

1.32. Information about 29mot-18

Content

MOT focuses on the knowledge and understanding of lifecycles in ICT. How will the fluctuate during economic situations and in several branches in time? What are tools to transform your business to another model or towards outsourcing.

What is the impact of low-code / no-code platforms like Blueriq and Outsystems?

Learning outcomes

Learning outcomes The student will be able to:

- Understand the needs for changes to business transformation
- Understand and apply the Nolan lifecycle model
- Understand the evolution of ICT Business integration
- Knows the competences of business integration and can apply them
- Understand the complexity of organizational change and relation to ICT
- Understand the impact on employees based on changes and transformations
- Able to understand and apply models like: TQM, Balanced Scorecard, Five Forces of Porter, Nolan-Norton Lifecycles, Creating value. INK.
- Understand and apply governance models
- Understand change management processes
- Understand the use of low code / no code platforms

Examination and grading

Examination

Cases during class, presentations and a final essay individual with a presentation

Tools allowed

Laptop and all handed over materials.

Resits and repairs

Every semester the course is given a possibility to repair is possible.

Grading

Final essay and presentation. During class attendence and collaboration

Teaching methods

7 classes of 4 hours

Resources

Sharepoint, pdf, books, handed over materials by the teacher.

1.33. Information about 29bia-18 Business IT alignment

Content

BIA (Architectural frameworks and Enterprise Architecture as Strategy)

Business and Information Technology Alignment

BIA is essentially the core of the Business IT training. Imagining the business on IT and matching it is the core of the training. At the strategic level, imaging is not a simple task and therefore use is often made of architectural methods and models. We continue to build on models that are used earlier in the curriculum, such as the reference models BISL, ITIL and ASL, but also ordering models such as the 7-S model from Mc Kinsey and DESTEP.

IT strategy and planning is not easy. By looking further ahead, the theory is much more detached from reality than within the models that focus more on the operation and organization of IT. BIA has a lot in common with the other courses in semester 7. Changing among architecture naturally has many points of contact with change management as it is dealt with in BPM. Many architectural models are also control models and therefore have many interfaces with the "Governance" models as they are treated at ITA.

- Architectural models, architecture methods, working under architecture, maturity models, strategy development models, testing information plan, Enterprise architecture as strategy

Learning outcomes

Learning outcomes. The student will be able to:

- Understand and apply current architectural models in practice from basic to more extended models
- Understand and apply the business goals which will be connected to the information provision and the information technology.
- Apply and prepare criteria for information plans which must be met
- Understand, apply an able to advice how strategy is developed in large service companies related to IT
- Apply the correct business models based on business criteria seen in a company, and deliver a proper advice

- Apply models like: the business model Canvas, to cases and real life situations Archimare and the Novius Architectural Framework

Examination and grading

Examination

Assignments during class and a final assignment for week 7

Tools allowed

Laptop and handed over materials by the teacher

Resits and repairs

Every semester the course is given a possibility to repair is possible.

Grading

Final assignment and active learning in the class

Teaching methods

7 classes of 4 hours

Resources

Sharepoint, and handed over by the teacher

1.34. Information about 29bpm-18

Content

Learning outcomes

Examination and grading

Examination

Tools allowed

Resits and repairs

Grading

Teaching methods

Resources

1.35. Information about Minor (free choice)

Content

Learning outcomes

Examination and grading

Examination

Tools allowed

Resits and repairs

Grading

Teaching methods

Resources

1.36. Information about Graduation Project

Content

During the last semester of the bachelor course Information & Communication Technology the student must show he can function at the level of a graduate by completing a graduation project. This project comprises 90 - 100 working days and is normally carried out in the business community.

In this brochure you will be informed about the preparation, the realization and the evaluation of the graduation project period

There are only 5 possible starting dates per year for your graduation: Four at the first or second week of every quarter, and a fifth possibility in week 5 of the spring semester (feb-july). Below, all start dates are indicated by A,B,C,D and E. Accompanying graduation proceedings dates are indicated by A, B, C, D and E. In some cases the proceedings can also take place 1 week sooner.

Failure to start or finish a graduation on the dates below will automatically mean the start or finish will be postponed to the next scheduled possibility. To be able to start in a quarter, the student needs to have an approved survey on Wednesday of week 8 of the previous quarter. (for instance: for starting in September, you will need to have an approved survey before Wednesday in week 8 of the April-July quarter)

Learning outcomes

- Student is able to apply all teached knowledge during the graduation project
- Student is capable of applying the DOT framework research model
- Student can critical assess the environment of the company and set frequirements and scope by interviewing business people
- Student has the capability to advice the company with recommendations
- Student is capable of working by him / her self
- Student takes an entrepeneural attitude
- Student is able to adapt to the business and codes of conduct and regulations in the company
- Student can finish his / her product / project within 90-100 days
- Student is able to present the results in a process way with enough underpinning of choices before a jury

Examination and grading

Examination

Graduation project > 5.5

Presentation before jury > 5.5

Process Report > 5.5

Tools allowed

Not applicable

Resits and repairs

Student can, after approval of the FHICT Examboard, repair either the presentation or the report but not the work. All parts must be 5.5 or higher. In case of repair a maximum mark of 6 is awarded.

Grading

Student is assessed at the company by the university tutor and the company tutor for the work contents and attitude at the company

Student is assessed and questioned during his / her defence presentation before the jury

Student is assessed on his / her process report

Teaching methods

University tutor visits the company and the student at least two times. One time between week 4-6 and one time to rehearse the final presentation and discuss the mark for the work at the company

Resources

all learned at Fontys ICT

2. Information about Specialisation ICT & Software Engineering

2.1. Information about OOD1 Object Oriented Development 1

Content

Subjects:

- UML Use-Cases
- UML Class diagrams
- UML sequence diagram
- Painting in C#
- Unit testing

Learning outcomes

The student will gather knowledge and competences about UML and implementing in an object-oriented way.

Examination and grading

Examination

written exam + practical

Tools allowed

nothing except your own brains

Resits and repairs

exams in the exam-weeks of the odd-numbered blocks

Grading

total of score for practica and exam, divided by 10 and rounded to the nearest integer

Teaching methods

theoretical lessons and practica

Resources

- lecture notes and ppt's
- Book UML Distilled, A brief guide to the standard Object Modelling Language, Third Edition, Martin Fowler,

2.2. Information about UX User Experience

Content

When using applications, user interfaces are the first things users see. From the perspective of the end user, the user interface IS the application!

Interaction between the UI and the user determines the usabilitiy of the application. Designing a UI is therefore a major skill ICT people should have.

Creating a UI encompasses executing a cycle in which you focus on 3 parts: find out what your users need (USER part), designing the UI (DESIGN part), testing if your design supports the need of your users (TEST part).

You will perform 3 of those cycles mentioned above in a group, in which you improve your UI design every time. These are called 3 increments. Furthermore, every student individually will explore a technique used in UI increments (exploration). Which technique, and how extensive it will be explored, will be planned at the beginning of every increment.

The group will apply the knowledge and skills from the explorations, and thereby improve the previous design. Also all goals for every increment are planned beforehand by the group.

Learning outcomes

During this course you will work on developing knowledge and skills.

- Setting up and executing a UI design process with a group.
- Incrementally designing a UI in a group.
- Personal research into techniques needed for UI design.
- Applying techniques during UI design.

Examination and grading

Examination

You will be coached by the teacher during every increment by receiving feedback, which you can use in future increments (feed forward). In the first increment, you only receive feedback and you will not

be assessed. The second and third increment will be assessed (by 2 teachers), and the assessment will be based on your group planning of the increment.

Tools allowed

All additional aids are allowed.

Resits and repairs

Failure to pass the course automatically means you will have to redo it completely.

Grading

The second and third increment will be assessed (by 2 teachers), and the assessment will be based on your group planning of the increment.

Teaching methods

general planning of increments, explorations and assessments

Week	Student (individual)	Student (team)	Teacher
1 - part a		Form a team Contact potential users	The Good, The Bad & The Ugly (30 min.)
1 - part b	Plan exploration 1 Work on exploration 1	Planning iteration 1 Work on iteration 1	UX: Why? What? How? (20 min.) Determining personal interest/goals/case Self organisation groups (3 students per group) Exploration
2 - part a	Work on exploration 1	Work on iteration 1	Feedback individual/subgroup/team
2 - part b	Hand in exploration 1	Work on iteration 1	Feedback individual/subgroup/team Done/NotDone+FB exploration 1
3 - part a		Work on iteration 1	Feedback exploration 1
3 - part b		Work on iteration 1 Handin iteration 1 (2 days before first lesson next week)	Feedback exploration 1
4 -	Plan	feedback	Feedback iteration 1 (1

part a	exploration 2	iteration 1	teacher, half class)
			Planning
		Work	iteration 2
		on	
		exploration 2	Work on
			iteration 2
4 - part b	Work on exploration 2	Work on iteration 2	Feedback iteration 1 (1 teacher, half class)
5 - part a	Hand in exploration 2	Work on iteration 2	Feedback exploration 2
5 - part b		Work on iteration 2	Feedback exploration 2
			Handin iteration 2 (2 days before first lesson next week)
6 - part a	Plan exploration 3	Feedback iteration 2	Feedback iteration 2 (2 teachers, half class)
			Planning
		Work	iteration 3
		on	
		exploration 3	Work on
			iteration 3
6 - part b	Plan exploration 3	Work on iteration 3	Feedback iteration 2 (2 teachers, half class)
		Work	
		on	
		exploration 3	
7 - part a	Hand in exploration 3	Work on iteration 3	Feedback exploration 3
7 - part b		Work on iteration 3	Feedback exploration 3
8	No exam.	Feedback iteration 3	Feedback and grade iteration 3 (2 teachers)

Resources

Online and other sources you have found yourself during explorations	
Online and other sources	

provided by teacher

Students will make an own choice for UI design software to use during this course.

Optional:

978-1-93435-675-3 Designed for use Lukas Mathis

2.3. Information about WEB2 Web Development 2

Content

The module focuses on front-end (HTML, CSS, JavaScript) development only. You will study state-of-the-art approaches of front-end web development. During this course students will build a front-end of a web application using a JavaScript framework called AngularJS. If the student successfully completes the course, she/he will be rewarded with 3 ECTS (84 hours of work).

Learning outcomes

The student will be able to:

- Understand the general design pattern MVC (Model, View, Controller).
- Provide insight about the need of a front-end framework in modern web development.
- Understand the main concepts defined in AngularJS framework (Directives, Module, Scope, Model, Expressions, Filter, Views, Controllers, etc.).
- Understand how AngularJS implements the MVC pattern and its variations such us MVW (Model, View, Whatever).
- Get to know the following front-end tools: Node.js, npm, Bower.
- Apply the knowledge acquired in a practical way by building a fully featured front-end web application.
- Use an API (Application Programming Interface) during implement of the front-end web application.

Required knowledge:

- GIT version control system
- HTML
- CSS
- JavaScript
- FIS2 (current IDE)

Examination and grading

Examination

The student needs to collaberate in a group to deliver AngularJS front-end web application with the required functionalities.

Tools allowed

N.A.

Resits and repairs

The student will need to redo the entrie course, when she/he needs to retake WEB2.

Grading

The grade of the Practical is the final grade of WEB2. For each deliverable (A, B, ..., F) you will get a literal “I”, “S”, “G”, “E” meaning “Insufficient”, “Sufficient”, “Good”, “Excellent” respectively. This is just an indication of your progress. The final grade will be given after you submit your web application in week 7.

There will be Go and No-Go sessions. This is determined by your teacher at any point in time during the course. A No-Go means your group can not proceed any further and must leave the course receiving 1 as final grade for this course. If your group misses a deadline you get “I” for that deliverable and your final grade will be decreased by 1 point. If your group misses more than one deadline you automatically get a No-Go.

Teaching methods

The approach used in this course is Learning-by-doing. Therefore, this course is compound of 1 theory hour and 3 practical hours per week, at school. Besides, the student will have to spend a minimum of 8 hours (self-study) in the matter per week. The theory lectures explain what a certain technology is, the purpose of it and the essentials on how to use it. It is the student task to go deeper and study further in detail such technologies so that she is bet ter prepared to tackle the issues that might appear during the practical assignment.

Furthermore, the student is expected to have the required knowledge previously stated to guarantee a smooth flow of the course, good code management and avoid falling behind.

The practical of this course is about building a front-end web application which is described later in Practical section (see “Practical” section). Since web development is mainly a collaborative process you will work in groups of 3 people.

Resources

This module uses the slides and internet resources:

Description	Link
AngularJS official website	https://angularjs.org/
AngularJS developer guide	https://docs.angularjs.org/guide
AngularJS API reference	https://docs.angularjs.org/api
AngularJS learning environment	http://www.w3schools.com/angular/

2.4. Information about WEB3 Web Development 3

Content

In this course we focus on back-end (PHP & MySQL) development. You will be applying the knowledge gained about front-end development during this course, but we will not teach you any new techniques. During this course you will be making use of a PHP Framework called Laravel. With this

framework you will create a multi-user web application with ‘dynamic’ content containing text and images (examples: reviews, recipes, life hacks, events, POI, etc.).

If you successfully complete the course you will be rewarded with 3ECTS (84 hours of work).

Learning outcomes

The student will be able to:

- Install a development environment for developing Laravel web applications;
- Explain how MVC is implemented in Laravel and how the model, view & controller work together;
- Use of Composer;
- Use of Artisan to enhance their Laravel developing experience (scaffolding, migration, cache clearing, etc);
- Create an 'environment configuration'-file for the respective environments
- Create a Controller with actions with appropriate responses;
- Use of Blade to create views, layouts & partial views;
- Judge which routes should be create in their Laravel web application;
- Create a model;
- Bind a model to a form in the view;
- Implement model validation by making use of Laravel Validator class;
- Use Eloquent to query the database for models;
- Use Eloquent to do CRUD actions for model to a database;
- Use migration & seeding files to populate a database with tables and initial content;
- Create authentication & authorization functionality by make using of the build-in features of Laravel;
- Create file upload functionalities in Laravel; Enabling a user to upload and view the uploaded file
- Make use of the 3rd party package to extend the functionalities of Laravel;

Create a simple RESTFUL API in Laravel;

Judge which request method should be used for the API;

Judge which mime type & status code should be returned by Laravel.

Examination and grading

Examination

The student needs to collaberate in a group to deliver Laravel Web Application with the required functionalities.

Tools allowed

N.A.

Resits and repairs

The student will need to redo the entrie course, when he/she needs to retake WEB3.

Previously earned badges are not applicable when redoing this course.

Grading

The grade will be based on the “Final commit in Master” of the deliverable on GIT Repo before week 8. The deliverable also needs to be bug free running on the Iris server. While the deliverable consist of group work the grade can be different per group member. The grade will be also based on the earned badges (max. of 4), group work and the way the features are implemented in the final version. This means your final grade will be based on following:

Description	Max grade
Final deliverable	6.0
Group work	3.0
Badges	1.0
Total	10.0

There will be Go and No-Go sessions in week 4. During this session your group needs to have implemented all the required practical features from week 1, 2 & 3. Your teacher will determine if it's implemented in a sufficient way to continue the course. A No-Go means your group cannot proceed any further and must leave the course receiving no grade for this course.

Teaching methods

Way of working

The approach used in this course is Learning-by-doing. Therefore, this course expects you to do self-study and the practical. Each week, during the first class, the topics of previous week will be explained by a student to other students.

Some topics will be marked, which will enable you to earn badges. Each week you can earn at most 1 badge and in total you can have at most 4 badges. In the last class of week 7 you can hand-in the badges. Each badge (you earned) you hand-in will award you with 0.25 points.

The teacher will function as a facilitator and will step in when needed. The teacher will also supply in-depth/background information which should enrich your understanding of the covered topics.

Furthermore, the student is expected to have the required knowledge previously stated to guarantee a smooth flow of the course, good code management and avoid falling behind.

The practical of this course is about building a Laravel web application which is described later in Practical section (see "Practical" section). Since web development is mainly a collaborative process, starting from week 2, you will work in groups of 2 people.

The practical

Starting from week 2 you will work in a group of 2 people. Your group will gradually build a multi user Laravel web application. The content of the web application is up to your group, but it is required you implement the features specified in de practical documents. The way these features are implemented is up to you/your group. Obviously this will influence your end grade.

There are enough features to work on, you will, at least, find a 2 features during all weeks except the first one which is about setting up your development environment individually. Ideally you should work in parallel (each member at least one feature), of course using GIT. The document also have a suggestion on how many people should work on a feature.

Resources

The resouces of this course exists out of:

- The slides;
- The official documentation from Laravel;
- The internet.

2.5. Information about EDB4 Databases 4

Content

Subjects:

- Design and implementation of (stored) procedures and cursors with PL/SQL;
- Implementation of assertions and triggers for the preservation of constraints;

Learning outcomes

The student must be able to design and implement small programs in PL/SQL using procedures, functions, exception handling, cursors and triggers.

Examination and grading

Examination

Laptop exam

Tools allowed

Laptop with SQL developer environment.

Resits and repairs

Next semester.

Grading

Mark for the exam is the final mark.

Teaching methods

Theory lessons + practical lessons.

Resources

Presentations and practical readers on Sharepoint

2.6. Information about SDS System Development for Software Engineering

Content

- Recall the **basic types of computer-based systems** that a systems analyst needs to address.
- Realize what the many **roles** of the systems analyst are.
- Comprehend the fundamentals of three development methodologies: **SDLC**, the **agile** approach, and **object-oriented** systems analysis and design.
-

Understand and be able to apply concepts of information systems, i.e.

- An overview of information systems.
- An introduction to information technology.
- The concept of “application.”
- Information systems as products.
- The business of developing information system products.
- Information system as the infrastructure of the business.

- The enterprise of software development.

Understand and be able to apply concepts of methodology, i.e.

- Fundamental concepts and building blocks of methodology.
- Benefits and risks of methodology.
- Software development methodologies.
- Modeling concepts and software development.
- Project management concepts and tools.

Understand and be able to apply the concepts of requirements gathering, i.e.

- Define requirements
- Requirements discovery
- Classifying requirements
- Techniques for eliciting requirements
- Managing requirements
- The case history of Walden Hospital, the main source for examples in this book

Understand and be able to apply the concepts of object-oriented analysis in the area of domain analysis, i.e.

- The three components of problem solving.
- The problem space vs. the solution space.
- Requirements vs. product specifications.
- Domains and their boundaries.
- Identifying domain concepts for analysis and modeling.
- Domain dictionaries and domain catalog.
- Identifying and organizing business rules.

Understand and be able to apply the concepts of object-oriented analysis in the area of behavioral modeling the basics of use cases, i.e.

- What use case modeling is and is not.
- The four components of a use case.
- The basic elements of use case diagram.
- How to transform concepts from domain analysis into use cases.
- Identifying prominent actors.
- Identifying major use cases.
- The context diagram.

Learning outcomes

- Understand the different system development methodologies
- Know when to use the different type of methodology based on criteria
- Student is able to apply theory in cases
- Student is able to describe the different methodologies
- Student can work in an agile project
- Student understand basic models of PM
- Student is able to choose between data gathering methods

- Student understands the problem space and solution space
- Student is able to model use cases
- Student has knowledge of all related terminology
- Student is able to gather requirements and model them for business
- Student is able to translate information gathering into models readable for business people

Examination and grading

Examination

- Groups: Assignments (Homework) and Presentations (WF: 0.4)
- Tests (WF: 0,6)

Tools allowed

None

Resits and repairs

Retake every semester. If passed for practical only exam retake

Grading

- You are actively present in all classes, which means:
- you arrive on time in class
- during assignments and presentations, you are active on the topic at hand,
- during theory you have an active attitude,
- you have prepared well before class and do your homework.

One class missed with valid reason is additional assignment.

Two classes or more missed. No grade for practical.

Teaching methods

6 classes of 4 hours theory and practical assignments / presentations

1 Guest lecture Agile / Scrum

Resources

Book: Ashrafi

- **Object Oriented Systems Analysis and Design**
- Noushin Ashrafi and Hessam Ashrafi
- First edition (New International Edition)
- Chapters: 1, 3, 4,5, 6 and

Book: to be selected (instead of Kendall)

2.7. Information about ProCp Project C-phase

Content

Project Core-Phase (ProCP) is a fourth semester module in which the students work in a group on a single complex application. The group is tasked with the development of a software solution that will facilitate a simulation of a real-life situation, allawing the end-user to improve.

The pre-requisited for ProCP are:

- Must have passed the propaedeutic phase.
- OOD1 needs to be at least a rounded 5 or higher.

When in doubt, the tutor can decide whether a student can **conditionally** participate in this module.

Learning outcomes

After successfully finishing this project you will be able to:

- Make a project plan for a software project
- Make a user requirements specification (URS)
- Make a design document consisting of a class diagram and a few sequence diagrams
- Implement an application conform a proper OO design for that application
- Make a test plan consisting of test cases
- Test an application conform the test plan
- Do a presentation

Examination and grading

Examination

The process and delievrables are assessed by the tutor per fase. At the end of the project, the group presents their solution and reflects on the project (presence is mandatory). After this presentation the tutor grade the group with a weighted average is taken.

The tutor can always decide to apply individual grades to (part of) the group.

Tools allowed

N.A.

Resits and repairs

When a student failed ProCP, he/she will have to restart the next semester. The student can participate in a new group.

Grading

The grading of ProCP is based on the following criteria:

1 Approval of project proposal & plan;
2 Active participation;
3 Presence during classes and project meetings;
4 Marked deliverables* submitted before deadline
5 Quality (sub-)deliverables.

*:See the Student workbook - Chapter 3: Week schedule.

If your group failed criteria 1 your work for ProCP is finished and the final mark will be an 1. For criteria 2 & 3: If you are not present for the first time: you will get a warning.

If you are not present for the second time: your work for ProCP is finished and your final mark is an 1. The first time your group failed criteria 4 a warning will be given. The second time missing a deadline results your work for ProCP is finished and your final mark is an 1.

If you pass criteria 1, 2, 3 & 4 then and only then your mark for the project will be determined by:

	Weight
Kick-off phase: • Project proposal and plan	Go or No-Go
Initial phase: • URS; • Work division	10%

report;
- Plan for iteration 1.

Iteration 1: 20%

- Updated URS;
- Design document;
- Test report;
- Working application(s) + source code;
- Updated work division report;
- Plan for iteration 2.

Iteration 2: 25%

- Updated URS;
- Updated design document;
- Updated test report;
- Updated working application(s) + source code;
- Updated work division report;
- Plan for iteration 3.

Iteration 3: 35%

- Final URS;
- Final design document;
- Final test report;
- Final version of working application(s) + source code.

End phase: 10%

- Process report and presentation

The final mark will be rounded to the nearest multiple of 0.5. For this module there is no final exam.

Teaching methods

The student will work in group formation on the project. A teacher tutors this process.

Resources

The resources for this course exists out of:

- Workbook ProCP;
- Theory covered in semester 1, 2 & 3 courses;
- The internet.

2.8. Information about OS1 Operating Systems 1

Content

This course is an introduction into operating system concepts, and also into VMWare, Linux, and the programming language Java. The most important topics of this course are:

- introduction in Linux, Java(FX) and Netbeans

- processes and threads

- event handling

- virtualization (running VMWare)

- scheduling algorithms

- synchronization issues in multithreading applications

Learning outcomes

The learning goals for this course:

- the student knows the most important Linux commands, and can develop and execute Java applications in a Linux environment running on VMWare

- the student can develop basic Java and JavaFX applications in Netbeans

- the student knows a couple of synchronization primitives in Java (wait, notify, synchronized) and can develop multiprocess and multithreaded applications in Java and JavaFX

- the student knows a couple of scheduling algorithms

Examination and grading

Examination

A written exam of 90 minutes at the end of the course.

Tools allowed

Nothing

Resits and repairs

There is no separate resit for the exam. Students who fail the exam can do the next regular exam in the next semester.

Grading

The final mark is based on the written exam at the end of the period. In this exam, no auxiliary items are allowed (no laptop, no papers, no books, etc).

The assignments of the practical part are a good preparation of the exam.

Teaching methods

7 classes of 2 x 2 hours per week (both theory and practical), and a written exam in week 8 or 9.

Resources

The course material (on the sharepoint) consists of:

- PPT slides for each week, and some Netbeans projects of demo's given in class.

- A lab manual that describes practical assignments for each week

- Class assignments for every week.

- Several Netbeans start projects for the practical assignments

- Old exams

For OS1, a canvas course exists that provides access to all course materials on the sharepoint

2.9. Information about DPR Design Patterns

Content

Using Design Patterns improves software development, because available solutions can be reused immediately. The application of Design Patterns improves the software quality, since the designs have proven themselves in earlier projects. In this course, some Design Patterns will be discussed and applied to solve specific problems. For the two/three design patterns selected by teacher for grading, the students will implement their solutions either in C# or Java programming language.

Learning outcomes

- a student can describe a Design Pattern and indicate its purpose
- a student can characterise some specific Design Patterns and implement them
- a student can indicate the strong and weak points of some specific Design Patterns
- a student can implement an application in which the strength of some specific Design Patterns is demonstrated

Examination and grading

Examination

The assignments are performed in groups of 2 students. Submission must be done in Canvas; students will be invited for the course.

For all the assignments submitted before the deadline students will get a feedback.

In week 7, teacher will announce which assignments should be submitted in week 8 for grading. For each of these assignments, a student will get a grade on a scale 1 to 10.

At the end of the course, in week 8 or 9, there will be a paper-based exam. For the exam, a student will get a grade on a scale 1 to 10.

All the assignments submitted for grading and the exam must be sufficient, mark >= 5.5.

Tools allowed

OO Programming languages like C# and Java, with their IDE's.

Class diagrams are written either on paper or using the tools **with no** reverse engeneering functionalities, e.g. UMLet.

Resits and repairs

No intermediate results are kept. In case of the resit, a student has to redo both the grading assignments and the exam.

Grading

Your final mark will be calculated the following way:

-

if any of the graded assignments or exam is insufficient, your final DPR mark will be the lowest mark of grades of the assignments and exam.

-

If all grades for assignments and exam are sufficient, your final DPR mark will be: (average mark of the marked assignments) x 0.3 + (final exam mark) x 0.7 = final mark

Teaching methods

- 5 weeks with classroom teaching and practical
- 1 week of self research

Resources

- Slides and Lab Manual on the portal
- Book “Head First Design Patterns”, O'Reilly Media , Eric T Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra, Oct 2004, ISBN:978-0-596-00712-6, ISBN 10:0-596-00712-4

2.10. Information about OS2 Operating Systems 2

Content

This course is a continuation of OS1. In the course, the following topics w.r.t. operating system are discussed:

- synchronization primitives in Java (use of wait, notify, synchronized, monitor classes, locks and conditions, countdown latch, cyclic barrier)

- the use of Java sockets for network communication

- deadlock, deadlock conditions, and the banker's algorithm to avoid deadlock

- multithreading in Java and the use of threadpools.

Learning outcomes

The learning goals for this course are:

- Students know the most important synchronization primitives in Java and can apply this knowledge in order to create a number of multithreaded applications that use shared data

- Students know the basics of socket communication and can apply the knowledge to modify an existing application into a client-server application that exploits socket communication.

- Students know the problem of deadlock, and how to avoid it using the banker's algorithm. They are also able to solve deadlock problems in a couple of Java applications.

Examination and grading

Examination

A written exam of 90 minutes at the end of the course.

Tools allowed

Nothing

Resits and repairs

There is no separate resit for the exam. Students who fail the exam can do the next regular exam in the next semester.

Grading

The final mark is based on the written exam at the end of the period. In this exam, no auxiliary items are allowed (no laptop, no papers, no books, etc).

The assignments of the practical part are a good preparation of the exam.

Teaching methods

7 classes of 2 x 2 hours per week (both theory and practical), and a written exam in week 8 or 9.

Resources

The course material (on the sharepoint) consists of:

- PPT slides for each week, and some Netbeans projects of demo's given in class.

- A lab manual that describes practical assignments for each week

- Class assignments for every week.

- Several Netbeans start projects for the practical assignments

- Old exams

For OS2, a canvas course exists that provides access to all course materials on the sharepoint

2.11. Information about PO/PD3 Professional Orientation and Personal Development 3

Content

The topics for this module are:

1 research-1: - SPA - systematic problem analysis.
2 research-2: main question -subquestions - Dot framework strategies and Methods applied to an internship project.
3 CV, how to contact a company
4 SWOT
5 finding a company
6 how to write a good projectplan

Prerequisites

- Knowledge about the DOT Research framework and the five strategies (FIELD, LIBRARY, WORKSHOP, LAB, SHOWROOM).
- Knowledge how to write a report.

- POPD1, POPD2 and PM : sufficient
- experience in a project.

Learning outcomes

Enhancing critical thinking skills, practicing techniques involving them and applying them in the context of internship and graduation.

Examination and grading

Examination

Every week there are deliverables in the form of an assignment. Also exercises done during class hours are considered deliverables.

Tools allowed

no end exam

Resits and repairs

One meeting can be missed, since one assignment can be repaired. Missing 2 meetings means failing the course.

Grading

The student will receive a score for each exercise and assignment.

Exercises will be scored done/not done. Assignments will be scored Insufficient/Sufficient/Good.

The end result is a combination of all exercises and assignments done.

All assignments need to scored sufficient or higher. If not: One assignment can be repaired in the week following the end of the course.

Not done exercises will lead to 0.5 of a point being subtracted of the final score of the assignments.

Assignments are graded on a linear scale going from 3 (all assignments insufficient), till 10 (all assignments good). All assignments sufficient will lead to a 6.

Teaching methods

individual and group assignments

Resources

powerpoint presentations

websites

youtube video's

DOT research documentation.

internship/graduation surveys

2.12. Information about CSA Client-Server Applications

Content

In this course you will learn how to make distributed Client-Service applications by using Windows Communication Framework (WCF).

Prerequisites:

Sufficient mark for modules PCS1, PCS2 and PCS3.

Learning outcomes

Learning outcomes of this course are:

- The student can use WCF operation contracts to connect client and service applications.
- The student can use programmatic and administrative configuration of service end points.
- The student can connect a client with a service via WFC proxy.
- The student can automatically generate a WCF proxy in the client application.
- The student can use WCF data contracts to transfer data between the client and the service.
- The student can make services with multiple operation contracts and end points.
- The student can make a service application that calls back the client application (two directions communication).
- The student can use publish-subscribe event-based communication between service and client.

Examination and grading

Examination

During the lessons you will get assignments for which you will get formative feedback (to which extent did you achieve the learning goals) and feed-forward (constructive guidance on how to improve) from the teacher.

One summative examination in which you individually apply the acquired skills in a new Client-Service case. The final mark is in the range 1 – 10 (1 – 5 is insufficient; 6 – 10 is sufficient).

Tools allowed

All additional aids stored localy on your laptop are allowed to use. It is not allowed to access material on the intenet or from other students.

Resits and repairs

Re-take is possible in the next semester.

Grading

Student gets a mark on the exam in the range 1 - 10.

Teaching methods

The course lasts 7 weeks. Each week there are 2 + 2 contact hours with the teacher. Each topic is shortly introduced by the teacher. For each topic a practical assignment is available, which helps students to practice and for whcih students can get feedback from the teacher.

Resources

- Slides and practical assignments on the SharePoint.
- Book “Programming WCF services” by Juval Löwy.

2.13. Information about Minor (free choice)

Content

Learning outcomes

Examination and grading

Examination

Tools allowed

Resits and repairs

Grading

Teaching methods

Resources

2.14. Information about Internship

Content

During the four year bachelor course Information & Communication Technology you will work as a trainee in a company (profit or non-profit).

This internship comprises 90-100 working days (planned in the second semester of the third year). Take enough time to prepare yourself!

In this brochure you will be informed about the preparation, the realization and the evaluation of the internship project period.

Learning outcomes

- Student is able to apply all teached knowledge during the graduation project
- Student is capable of applying the DOT framework research model
- Student can critical assess the environment of the company and set frequirements and scope by interviewing business people
- Student has the capability to advice the company with recommendations
- Student is capable of working by him / her self
- Student takes an entrepeneural attitude
- Student is able to adapt to the business and codes of conduct and regulations in the company

Student can finish his / her product / project within 90-100 days

Examination and grading

Examination

internship project > 5.5 (40%)

Presentation before university tutor and company tutor > 5.5 (20%)

Process Report > 5.5 (40%)

Tools allowed

Not applicable

Resits and repairs

Student can, after approval of the FHICT Examboard, repair either the presentation or the report but not the work. All parts must be 5.5 or higher. In case of repair a maximum mark of 6 is awarded.

Grading

Student is assessed at the company by the university tutor and the company tutor for the work contents and attitude at the company

Student is assessed and questioned during his / her presentation before the university tutor and the company tutor

Student is assessed on his / her process report

Teaching methods

University tutor visits the company and the student at least two times. One time between week 4-6 and one time to rehearse the final presentation and discuss the marks

Resources

all learned at Fontys ICT

2.15. Information about ANDR1 Android 1

Content

In this course, you will learn how to design and create basic Android apps in Android Studio.

Prerequisites

A good knowledge of an Object Oriented language (like C#) is required, as well as the ability to use a regular IDE (like Visual Studio).

Learning outcomes

Learning outcomes of this course are:

- The student can design an Android app according to the Material Design.
- The student can create Activities with Views and Layout.
- The students understands the steps in the lifecycle of an Activity.
- The student can create event handlers, and knows how to use Intents.
- The student can create and use Fragments.
- The student can read and interpret JSON data.
- The student can apply Android Manager classes.

Examination and grading

Examination

During the weeks, there are assignments for which students can get formative feedback and feed-forward (constructive guidance on how to improve) from the teacher. These assignments are not mandatory.

Each student must implement a basic app, according to some given requirements.

There is a written summative exam afterwards, in which the learning objectives are tested.

Tools allowed

Everything on paper.

Resits and repairs

Re-take is possible in the next semester.

Grading

The app that was created must be demonstrated and explained individually before the written exam. If this is done sufficiently, the final grade is the grade for the written exam. If the app is not demonstrated or explained sufficiently, the final grade is 1.

Teaching methods

The course lasts 7 weeks. Each week there are 2 + 2 contact hours with the teacher. Each topic is introduced by the teacher. For each topic a practical assignment is available, which helps students to practice and for which students can get feedback from the teacher. The student can work on the required app.

Resources

Slides and practical assignments on SharePoint, accessible through Canvas.

Android documentation and tutorials.

2.16. Information about ANDR2 Android 2

Content

In this course, you will do a project in which you design, create, test and deploy an advanced Android app in Android Studio.

Prerequisites

ANDR1 must have been finished with at least 5.

Learning outcomes

Learning outcomes of this course are:

- The student can use Android Location service.
- The student can use a backend server (for example FireBase).
- The students can use multithreading in Android.
- The student can make unit tests and instrumented tests in Android Studio..
- The student can deploy an Android app to user(s)

Examination and grading

Examination

A group of students (typically 4) will design, build, test and deploy an Android of their choosing, following some specific requirements given.

Tools allowed

Anything

Resits and repairs

Re-take is possible in the next semester.

Grading

Students will be graded individually, depending on the project result and peer assessments. A grading scheme is provided as guideline.

Teaching methods

The course lasts 7 weeks. Each topic is introduced by the teacher, and can be applied in the project. Remaining time is used to work on the project.

Resources

Slides and practical assignments on SharePoint, accessible through Canvas.

Android documentation and tutorials.

2.17. Information about ITOPS IT Operations

Content

Subjects:

- IT service management processes according to ITIL v3 (grouped into service strategy,
- service design, service transition, service operation, and continuous improvement)
- Process approach of an organization; 3 layer organization: Strategic, Tactical,
- Operational
- Quality management; Advising.
- Preparation for the official ITIL v3 Foundation exam

Learning outcomes

- The student has an overview in the field of IT service management, and has an understanding of different methodologies to manage an IT operations department ;
- The student gains knowledge of the ITIL methodology;
- The student understands and can advise how to set up and improve IT services (in the form of a report and a presentation);
- The student is prepared for the official ITIL Foundation exam.

Examination and grading

Examination

- written exam (50%)
- case advisory report and 2 presentations (50%)

Tools allowed

Geen.

Resits and repairs

Volgende semester is er herkansing.

Grading

Your grade is built up based upon group activities (with group of 4-6 studentes) and individual activities. For students who show higher or lower performance in their group thant the group average,

a higher or lower grade will be given. For each assignment or test you can get a score between 1 and 10 (with 10 highest grade).

- Group activities:
  - Do research to an IT operations/IT Service management method and present on it. (25 % of final grade)
  - Based upon a given case, develop an advisory report and present it. (25% of final grade)
- Individual activities:
  - 4 small tests in weeks 2, 3, 5 en 6: the tests will be held at the start of the lectures and will take approximately 10 minutes. For each sufficient test you can score 0.2 bonus points for the final test. If all tests are scored sufficiently you will receive an extra bonus of 0.2 points. This will lead to a maximum total of 1 bonuspoint on the final test.
  - Final test: At the end of the ITOPS course, you will get a test which is a combination of multiple choice questions and open questions. The focus of this test is ITIL v3. (50% of final grade).

- If the result for the final test is insufficient (lower than 5.5), the result of the final test will define your final grade.
- Attendance: this is compulsory for each lecture and for each group activity. You are allowed to be absent **once.** However, you need to state official reasons for doing so.

Teaching methods

A true understanding of IT service management requires that you practice yourself with setting up IT service management. Therefore, doing own research, working on cases and presenting your advise is an important part of this course. Every new concept of IT service management and ITIL is introduced as a lecture.

- A weekly lecture of maximum 1 hour in which an introduction is given on specific parts of the matter. You can further elaborate on the topic with the textbook and by working in small project groups.
- In the weeks 2, 3, 5 and 6 each lecture will start with a small test. The individual scores have a weight in the final result. (See study guide how the final grade is defined).
- Per group of 4-6 students, work on a given case that leads to an advise report and a presentation about your findings and advise.
- Per group of 4-6 students, choose a topic related to IT operations, do research about it and present to the whole class.
- Exam in which you show your understanding of IT operations (partly multiple choice, partly open questions).

Resources

- Foundation of ITIL v3, Jan Van Bon, et al. Van Haren Publishing. ISBN 978 90 8753 057 0
- Sharepoint environment: ITOPS reader, case description, slides

2.18. Information about EBUS E-Business

Content

- *E-tech: the backgrounds about the function and structure of networks and servers in general and specially of the internet.*
- *E-com: the economical, organizational and commercial consequences of doing business over the internet (e-commerce).*
- *E-prog: basic knowledge about the most used languages and tools for building a website and a web shop.*
- *E-project: a project in which a web shop is being build.*

Learning outcomes

- *Technical knowledge about the function and structure of networks and servers in general and specially of the internet.*
- *Knowledge about the economical, organizational and commercial consequences of doing business over the internet (e-commerce).*
- *Knowledge about the most used languages and tools for building a website and a web shop.*

Examination and grading

Examination

- *E-tech: exam*
- *E-ecom: case study and presentation on an e-commerce related subject.*

E-project: a ready made and functional web shop

Tools allowed

none

Resits and repairs

retake every semester

Grading

Review of exam and case study. 2 Separate marks

Teaching methods

7 classes of 4 hours

Resources

- *lecturer notes and ppt's,.*
- *books :*

2.19. Information about BEC1 Business Economics 1

Content

Topics that are discussed during the module

- Introduction to financial reports
- Introduction to financial ratio analysis
- Setup a bookkeeping system and calculate profit and equity. Further calculate costs, expenditures and incomes.
- Financial accounting scenario: ledger, journals, day book, subsidiaries, computing
- Special entries: accrual vs cash system, error bookings, interest bookings, private accounts, funds in transfer
-

Financial Accounting.

- Balance Sheet
- Income statement
- Retained earinings

- Basic GAAP rules
- Fixed assets vs. current assets
- Drawings account
- Cash transfers
- Accrued cost
- Annual Closure (Trial Balance)
- Balanced score card

Learning outcomes

Understand the importance of financial reports
understand the different types of financial ratios
Able to understand the numbers of the financial ratios
Able to compare companies of different size based on ratio analysis
Able to setup a basic bookkeeping system
Able to understand Trial balances, balances, profit & loss statements
understand taxation systems

Examination and grading

Examination

- Group assignments which have been assessed by individual contribution and individual portfolios.(40%) and small test in week 5
- Written exam (60%) in week 8

Tools allowed

pen and calculator

Resits and repairs

Next semester retake

Grading

- Individual feedback on assignments
- Group feedback on assignments and presentations
- Individual feedback on small test in week 5
- Weeks of presentations are mandatory

Teaching methods

7 classes of 4 hour with theory, presentations, assignments

Resources

-

From bookboon

- The accounting Cycle (PDF file)
- International financial accounting. (PDF file)
- Provided slides

2.20. Information about BEC2 Business Economics 2

Content

- Introduction to financial reports
- Introduction to financial ratio analysis
- Setup a bookkeeping system and calculate profit and equity. Further calculate costs, expenditures and incomes.
- Financial accounting scenario: ledger, journals, day book, subsidiaries, computing
- Special entries: accrual vs cash system, error bookings, interest bookings, private accounts, funds in transfer

Learning outcomes

- Understand the importance of financial reports
- understand the different types of financial ratios
- Able to understand the numbers of the financial ratios
- Able to compare companies of different size based on ratio analysis
- Able to setup a basic bookkeeping system
- Able to understand Trial balances, balances, profit & loss statements
- understand taxation systems

Examination and grading

Examination

- Group assignments which have been assessed by individual contribution and individual portfolios.(40%) and small test in week 5
- Written exam (60%) in week 8

Tools allowed

pen and calculator

Resits and repairs

Next semester redo of the exam or retake of the course in case practical failed

Grading

Presentations / Assignments (40%)

Exam written week 8 (60%)

Teaching methods

7 Classes of 4 hours

Theory, assignments, feedback, presentations

Resources

- The Accounting Cycle; Authors Larry M. Walthr; Christopher J. Skousen; ISBN 978-87-7681-486-1
- Slides and practical assignments on the SharePoint.
- Internet sources
- The essence of financial accounting - Leslie Chadwick - ISBN: 0-13-356510-6

2.21. Information about SePr Secure Programming

Content

Subjects:

- The Secure Development Lifecycle (SDL).
- Overview Hacking threats, hacking techniques, possible impact.
- Risk and security threat analysis for software systems. (STRIDE, Attack Trees, misuse cases).
- Analysis and application of Software Security Principles.
- Analysis programming language and environment on vulnerabilities.
- Analysis programming language and environment on available security functionality.

Learning outcomes

- Learn what hacking techniques are used by hackers, and learn how these techniques work.
- Being able to analyse and apply a secure software development process (risk analysis, secure design, coding, testing).
- Have insight in the most essential security related coding problems and being able to find and apply solutions in software systems.

Examination and grading

Examination

- peer learning report on web hacking techniques assessment
- SDL presentation assessment
- group work report assessment
- final presentation assessment

Tools allowed

N/A

Resits and repairs

In the next semester

Grading

Final grade is 65% of peer learning hacking techniques report assessment and 35% of the group work assessment

Teaching methods

lectures and practicals

Resources

lectures (theory) and practicals (instructions)

2.22. Information about IPV Image Processing Vision

Content

This course is an introduction of the basics/advanced techniques of **image processing** and **computer vision.**

Prerequisites

Students need to pass all the following courses:

- Programming courses(including OS1, OS2 courses) from the first and second year
- Math2 course

Learning outcomes

- knows the basics of image processing (pixels, color representation, color depth).
- knows what a convolution filter is, and can implement such a filter on the pixel level.
- knows when to use some basic filters: mean, median, Gaussian, Sobel.
- knows how to apply Hough line detection.
- knows the k-nearest technique and can apply it to recognize digits.
- knows the Haar cascade technique and can apply it to recognize certain objects, like face, mouth, eye.

Examination and grading

Examination

Assessment

Week 1 is the registration for this course. You must be present and register yourself if you would like to select this course.

The final grade of IPV course is based on **two assignments (week 3 and 4)**:

- These two assignments (week 3 and 4) must be submitted on CANVAS on time respectively.(**deadlines are strict! If you miss the deadline, you get 0 for that assignment**).
- For each of these assignments (week 3 and 4) students will get a grade:

- If **your .mlx file** in the assignment does not compile properly or cannot run on the computer of the teacher (e.g., due to missing images, syntax errors, etc.) , then your assignment will not be checked, and your grade for that assignment will be 1.
- You get a grade 1-10 for each assignment.
- Final grade:
  - If both assignment grades >= 6, final grade = average of two assignments
  - If **not** all assignment grades >= 6, final grade = minimal assignment grade
- The rest assignments in weeks 1,2, 6 are for students to practise.
- **Bonus(optional)->1 point added to final grade**
- Students can gain bonus point by submitting week6 assignment(Computer Vision part) on CANVAS on time.
- Students should be able to demo the assignment to the teacher, and explain correctly and clearly what they have done.

Tools allowed

NA

Resits and repairs

NA

Grading

See 3.2

Teaching methods

Theory classes in which the main concepts are explained, using slides and example code.

Practical classes in which the students can work on the assignments.

Resources

Slides, assignments, some background material.

2.23. Information about SOT Service Oriented Techniques

Content

In this course you will learn how to make (web) services and clients. The course covers two techniques in programming language Java: (1) RESTfull service applications and RESTfull client applications for synchronous communcation; and (2) Java Messaging Service for asynchronous request-reply communication between two applications.

Learning outcomes

Learning outcomes of this course are:

- **Student can describe the differences between RESTfull and JMS as synchronous/asynchronous techniques.**
- **REST learning outcomes:**
  - Student can make a RESTfull service and client in Java.
  - Student can deploy a RESTfull service via an HTTP server.
  - Student can deploy a RESTfull service on Tomcat.
  - The student can use JSON serialization.
- **Java Messaging Service learning outcomes:**
  - Student can make Java applications which send and receive JMS messages.
  - Student can use both JMS Queues and Topics for exchanging messages.

- Student can make a request-reply type of communication with JMS between two Java applications.

Entry requirements

Successful completion of:

- year 1 of software engineering studies, and
- year 2 of software engineering studies, and
- year 3 of the software engineering studies, and
- the Internship.

Examination and grading

Examination

During the lessons you will get assignments for which you will get formative feedback (to which extent did you achieve the learning goals) and feed-forward (constructive guidance on how to improve) from the teacher.

Students make three assignments at home, these assignemnts are for summative assessment:

1 REST assignment - In this assignment students make a RESTfull web service and client application(s).

- JMS request-reply assignment – In this assignment students use JMS to realize asynchronous request-reply communication between two Java applications.

Tools allowed

All additional aids are allowed.

Resits and repairs

Re-take is possible in the next semester.

Grading

For each of the two assignments (REST and JMS) a grade in the range 1 – 10 is given.

Final mark for SOT (**int** sot_grade) is in range 1 – 10 and it is calculated as folows based on "**int** rest_grade" and "**int** jms_grade":

if (rest_grade >= 6 && jms_grade >= 6){ **float** average_grade = (rest_grade + jms_grade) / 2; sot_grade = Math.*round*(average_grade); } **else** { sot_grade = Math.*min*(rest_grade, jms_grade); }

Teaching methods

The course lasts 7 weeks. Each week there are 2+ 2 contact hours with teh teacher. Each topic is shortly introduced by the teacher. For each topic a practical assignment is available, which helps students to practice and for which they can get feedback from the teacher.

Resources

- Slides and practical assignments on the SharePoint.
- Java EE, Java Messaging Service, JAX-RS.
- Apache Tomcat, Apache ActiveMQ.

2.24. Information about SAI Software Application Integration

Content

In order to execute one business process, users often need to use many different “business” software applications. This is why these “business” applications should be connected with each other through some kind of “middleware” application. The “middleware” integrates all “busi” applications by passing data between applications, performing necessary data transformation and routing. In general, “middleware” application can be made by (1) making a custom software application, or (2) using a generic tool like MuleSoft, JBoss, etc.

In this course you will learn how to use Enterprise Integration Patterns in order to make a custom middleware application which integrates several “business” applications.

Prerequisites:

Pre-knowledge of:

- Java,
- Java Messaging Service with Enterprise Application Integration patterns Return Address and Correlation Identifier,
- Java SOAP client, and
- Java REST client.

Learning outcomes

After this course, the student can:

- Apply and Implement Broker pattern.
- Apply and Implement Chained Gateways pattern.
- Apply and Implement Content Router and Content Enricher patterns.
- Apply and Implement Scatter-Gather pattern.

Entry requirements

Successful completion of:

- year 1 of software engineering studies, and
- year 2 of software engineering studies, and
- year 3 of the software engineering studies, and
- the Internship, and
- module Service Oriented Techniques (SOT).

Examination and grading

Examination

During the lessons you will get assignments for which you will get formative feedback (to which extent did you achieve the learning goals) and feed-forward (constructive guidance on how to improve) from the teacher.

For summative assessment students implement integration of a real-life example. Approximately 5 “business” applications are delivered, and students implement the integration based on a given business case.

Tools allowed

All additional aids are allowed.

Resits and repairs

Re-take is possible in the next semester.

Grading

Students implement integration of a real-life case example assignment. This assignment is presented to students in week 6. Approximately 5 “business” applications are delivered, and students implement the integration based on a given business case. The final grade is determined based on the following matrix:

	6	7	8	9	10
The system works correctly with one "bank" application	x	x	x	x	x
Message Broker	x	x	x	x	x
Correlation Identifier	x	x	x	x	x
Return Address	x	x	x	x	x
Content-Based Router		x	x	x	x
Content Enricher		x	x	x	x
Messaging Gateway			x	x	x
Chained Gateways			x	x	x
The system works correctly with three "bank" applications				x	x
Recipient List				x	x
Aggregator				x	x
Scatter-Gather				x	x
Flexible evaluation of "bank" rules (e.g., use of Jeval library).					x

Final marks

Teaching methods

The course lasts 7 weeks. Each week there are 2 hours of lecture and 2 hours of practical. A lecture consists of a short presentation of main topics followed by a demonstration by the teacher. For each topic a practical assignment is available, which helps students to practice. In week 6 the final assignment is presented (for final assessment).

Resources

Book: Enterprise Integration Patterns : Designing, Building, and Deploying Messaging Solutions, by Gregor Hohpe, Bobby Woolf, ISBN 0321200683, Addison-Wesley, 2004

Technologies: Java EE, Java Messaging Service, JAX-WS, JAX-RS.

2.25. Information about TLA Applied Linear Algebra

Content

This course covers various aspects regarding applied linear algebra for application in computer graphics:

- distances between points, lines and planes
- angles between lines and planes
- left- and right-rotating coordinates systems
- converting a coordinates system
- representing a 3D-world on the screen
- rotations and translations in R3
- matrix calculations
- painter's algorithm
- parallel light and point light

Learning outcomes

The student can do calculations on:

- distances between points, lines and planes
- angles between lines and planes
- left- and right-rotating coordinates systems
- converting a coordinates system
- representing a 3D-world on the screen
- rotations and translations in R3
- matrix calculations
- painter's algorithm
- parallel light and point light

Examination and grading

Examination

written exam

Tools allowed

simple calculator + Derive

Resits and repairs

next semester

Grading

final grade is grade of written exam

Teaching methods

lectures with theory and practical

Resources

slides, reader, exercises

Derive

2.26. Information about TCI Testing and Continuous Integration

Content

In this course you will learn and apply different types of testing techniques by using a variety of JAVA frameworks. Also, best code practices quality insurance will be discussed through a process of continous integration.

Prerequisites

- Knowledge of the JAVA language;
- Knowledge of Client-server architecture (CSA) and web responses (WEB1-2-3);
- Knowledge of a JAVA web application technique (RESTful or SOAP using either POJO and Apache Tom Cat or Spring);
- Math 3 (graphs theory and search algorithms);
- Knowledge of GIT (pulling, pushing and merging);

Learning outcomes

The learning outcomes of this course are:

- The students know how to write unit test that make use of mockups, using Mockito Framework, to cover the application project they work on during the course;
- The students apply use of code styles in the assignment project;
- The students learn how to make their code decoupled by using building tools (Gradle) and manage dependencies;
- The students are capable of unit tests, using JUnit framework, that cover an application project they work on during the course;
- The student can perform tests, other than unit tests, on his software (for instance: make use of JMeter to run stress tests on his application and create a report with it);
- The student learn about code coverage and perform code analysis making use of Intellj IDEA and applying it on the project they will work on.;
- The student practices collaboration by making pull requests through online collaboration tools and fixing them via GIT clients;
- The student knows how to install and/or use a Jenkins continuous integration environment for JAVA application development.

Examination and grading

Examination

During the lectures the student is prepared to face the practical assignments and receives formative feedback (to which extent did you achieve the learning goals) and feed-forward (constructive guidance on how to improve) from the teacher.

The final summative examination is conducted at the end and aims to verify the acquired skills. The test consists in a check that considers the learning goals:

- Application development
- Unit Testing
- Stress Testing
- Code style
- Documentation and conventionts

The final mark in in the range 1 – 100 (translated then into the related two digits value). For a complete list of the assessment criteria, check the teachers' manual.

Tools allowed

All additional aids stored locally on your laptop are allowed to use. Also web resources are allowed but plagiarism is **not** allowed. The code written by somebody else **must receive acknowledgment.**

Resits and repairs

The re-take is allowed in the following semester.

Grading

Student gets a mark on the examination in the range 1 – 10. The value is rounded up to half a point.

Teaching methods

The course lasts 7 weeks. Each week there are 2 + 2 contact hours with the teacher. Each topic is shortly introduced by the teacher and throughly explained preceding the practical hours.

For each topic a practical assignment is made available to practice, which helps students to practice and for which students can get feedback from the teacher. These practical assignment are not mandatory but they represent a good tutorial before start.

Resources

- Slides and practical assignments on the SharePoint.
- Book “Practical Unit Testing with JUnit and Mockito” by Tomasz Kaczanowski.
- Each practical includes a part of suggested reading and tutorials.

2.27. Information about QP Quality Principles

Content

Introduction

Topics that are discussed during the module

- Introduction Quality Management
- Perspectives of Quality Principles
- Software: CMMI, SPI,
- Production: Kaizen, Lean, Six Sigma
- Financial: IT Auditing, IFRS, BASEL II
- Compliancy: Sas-70, SOX, Laws, Pharma. Regulations
- Controls: Porter, Kaplan, Balanced Scorecard, Deming
- Administration: ISO 9001:2000
- Security: ISO 27001

- Ethics: Codes of conduct, Declaration on oath, anti-bribary acts
- Risk Management

Learning outcomes

- Understand the importance of quality systems
- Student is capable of applying quality models on real cases
- Student has developed a critical attitude regarding quality systems and the achievableness of the implementation
- Student works analytical and has the ability to advice management based on research
- Student has developed skills in writing an essay
- Student has developed a critical view environment and is able to judge impacts of measures and regulations

Examination and grading

Examination

- Groups: Assignments (Homework), Presentations and individual essay (WF: 0.7)
- Written exam (WF: 0,3)

Tools allowed

- Books
- Laptop
- Internet

Resits and repairs

Retake every semester of the written exam

Grading

- **Assessment of Tests, Assignments and Presentations**

At Fontys ICT we want to stimulate that students are very active in their classes.

Group Assignments and Presentations and individual essay are graded with 70% of the final score if:

You are actively present in all classes, which means:

- you arrive on time in class
- during assignments and presentations, you are active on the topic at hand,
- during theory you have an active attitude,

you have prepared well before class and do your homework.

You’re individual essay contains:

- 3 chosen topics applied to the company chosen
- In Euphorus <10% defects and plagiarism
- Has your critical view on the topics
- Contains a strong stakeholder analysis and quadrant
- Contains a worked out Porter Value Chain and balanced scorecard applied to your company with critical underpinning
- Comply with DOT framework research model
- APA References used
- Your point of view as CEO of the company with “out of the box” solutions.

Teaching methods

7 classes of 4 hours of which one class a Guest Lecture about Lean Six Sigma / Kaizen

Resources

Resources on Sharepoint published

2.28. Information about DWH Data warehousing

Content

- Understanding Data Warehousing: the new paradigm specifically intended to provide vital strategic information for the business managers and analysts.
- Understanding/investigating the radical changes for the information technology departments.
- Understanding impact of data warehousing on IT professionals and business users.
- Study topics: planning, requirements, architecture, infrastructure, design, data preparation, information delivery, deployment, and maintenance of a data warehouse.
- Big Data / Data analysis guest lecture
- Practical test implementation of DWH with Oracle and ETL tools

Learning outcomes

- Student is able to apply theory to practical cases
- Student can model the right architecture based on certain criteria
- Student has knowledge of required infrastructure
- Student understands the business impact of DWH's
- Student can advice companies for an apropriate architecture

Examination and grading

Examination

•Presentations + cases 40% (mark >=5)

•Exam 60% (chapters 1-13, 15 and 17 book DWH Fundamentals) (mark >=5)

•Total mark >=5.5 to pass!

Tools allowed

•book ‘Data Warehousing Fundamentals’ by Paulraj Ponniah

•printout sheets lectures DWH

•hand written notes

•Laptop + internet

•No mobile phones! No Facebook , messenger etc.

Resits and repairs

Retake every semester

Grading

Review of assignments

Review and feedback of presentations

Review and feedback written exam

Teaching methods

7 weekly classes of 4 hours of which 1 guest lecture "Big Data and BI Analysis / Machine Learning

Resources

•lecture sheets (partially based on book)

•advise:

book ‘Data Warehousing Fundamentals’ by Paulraj Ponniah, second edition isbn 978-0-470-46207-2

2.29. Information about ALE1 Automata and Logic for S 1

Content

This course is about the handling of logical propositions: read a proposition (in ASCII), make an internal tree representation, calculate the truth table, convert into the disjuctive normal form

Learning outcomes

Learning outcomes of this course are:

- Handling of logic propositions
- UML design
- algorithms & datastructures
- testing

Examination and grading

Examination

During the lessons you will get assignments for which you will get formative feedback (to which extent did you achieve the learning goals) and feed-forward (constructive guidance on how to improve) from the teacher.

Tools allowed

All additional aids are allowed.

Resits and repairs

Re-take is possible in the next semester.

Grading

To pass this course (grade 6), all assignments are implemented and work properly with an easy-to-use GUI (even on another machine (in particular: the lecturer's)).

For higher grades: incorporate the following aspects:

- good software design (classes, interfaces, SOLID principles, Design Patterns, …)
- clear documentation of your actual design and your design decisions
- proof of the robustness of your code (thorough test cases, code analysis, code coverage)
- robust recovery for incorrect user input
- other smart inventions and spectacular new features

Teaching methods

The course lasts 7 weeks. Each week there are 2 contact hours with the teacher. Each topic is shortly introduced by the teacher. For each topic a practical assignment is available, which helps students to practice and for which they can get feedback from the teacher.

Resources

- Slides and practical assignments on the SharePoint.
- common programming langauges like C# and Java

2.30. Information about ALE2 Automata and Logic for S 2

Content

This course is about the handling of state machines: read a finite state machine (in ASCII), make an internal representation, determine if strings belong to its language, convert it into a non-deterministic automaton, read regular expressions and read push down automata.

Learning outcomes

Learning outcomes of this course are:

- Handling of Finite State Machines
- Push Down Automata
- UML design
- algorithms & datastructures
- testing

Examination and grading

Examination

During the lessons you will get assignments for which you will get formative feedback (to which extent did you achieve the learning goals) and feed-forward (constructive guidance on how to improve) from the teacher.

Tools allowed

All additional aids are allowed.

Resits and repairs

Re-take is possible in the next semester.

Grading

To pass this course (grade 6), all assignments are implemented and work properly with an easy-to-use GUI (even on another machine (in particular: the lecturer's)).

For higher grades: incorporate the following aspects:

- good software design (classes, interfaces, SOLID principles, Design Patterns, …)
- clear documentation of your actual design and your design decisions
- proof of the robustness of your code (thorough test cases, code analysis, code coverage)
- robust recovery for incorrect user input
- other smart inventions and spectacular new features

Teaching methods

The course lasts 7 weeks. Each week there are 2 contact hours with teh teacher. Each topic is shortly introduced by the teacher. For each topic a practical assignment is available, which helps students to practice and for which they can get feedback from the teacher.

Resources

- slides and lab manual available on Sharepoint
- all online resources for common languages like C# and Java

2.31. Information about SD3

Content

Application of SD1, 2 and 3 in a functional design of information systems

Learning outcomes

- Student can apply theory into a functional design
- Student is able to use elicitation techniques to gather data and model UML
-

Examination and grading

Examination

- Written exam (60%)
- Theoretical and practical assignments and presentations (40%)

Tools allowed

None

Resits and repairs

Retake every semester

Grading

Review and feedback of presentations

feedback and review of written exam

Teaching methods

7 weekly classes of 4 hours

Resources

- **Book:** Object Oriented Systems Analysis and Design, Ashrafi & Ashrafi 1e 2009 ISBN: 978-1-29203-960-2

2.32. Information about SD4

Content

- The current collection of SOA books and articles is rich on high-level theory but light on practical advice. At the other end of the spectrum are the Web Services books that concentrate on APIs and programming, but gloss over the architecture.
- The Promise of SOA
- SOA — Architecture Fundamentals
- Business Architecture; BPM and SOA
- Service Context and Common Semantics
- Identify and Specify Services
- Design Service Interfaces and Service Implementations

Learning outcomes

- Student is able to understand the differences between services
- Student is able to model a process in BPM tool Bonitasoft
- Student understands the theory of a SOA architecture
- Student is able to advice a company based on infrastructure landscape with SOA components
- Students has knowledge of SOAP, REST and message brokers

Examination and grading

Examination

- Written exam (60%)
- Theoretical and practical assignments and presentations (40%)

Tools allowed

none

Resits and repairs

Retake every semester

Grading

Review of presentations and feedback

Solutions to homework cases

Teaching methods

7 classes of 4 hours

Resources

- **Book:** Object Oriented Systems Analysis and Design, Ashrafi & Ashrafi 1e 2009 ISBN: 978-1-29203-960-2
-

Web Services & SOA – Principles and Technology

o Michael P. Papazoglou: Web Services & SOA Chapters: Comprehensive case study, 1, 4, 5, 8, 15 and 16.1-16.9

- Lectures, powerpoints teacher

2.33. Information about CRY Crypography

Content

- whole number theory (to really understand RSA)
- symmetric key cryptography (for example DES or AES)
- public key cryptography (for example RSA)

Learning outcomes

- whole number theory (to really understand RSA)
- symmetric key cryptography (for example DES or AES)
- public key cryptography (for example RSA)

Examination and grading

Examination

2 practical assignments + written exam

Tools allowed

simple calculator

Resits and repairs

next semester

Grading

weighted average of practicals and written exam

Teaching methods

lectures + practical

Resources

slides + LabManual on the intranet

2.34. Information about ProEp Project E-phase

Content

Apply skills acquired during the whole study and apply them in one big project where a distributed system is created.

The main goals of this project are:

Have students use their inquisitive nature and critical thinking skills to define, execute and evaluate a justifiable project.

Prepare for the graduation project.

Students execute the project by developing a distributed application, in which the group is responsible both for performing software development and implementation of it on behalf of a client. Choices made need to be justified. Milestones and deadlines marking the progress of the project will be set in general terms.

Start on a high-level design (conceptual) and start implementing the biggest risks. Learn from previous iterations to speed up development.

Focus is on delivering a working system while keeping documentation in mind. Justifying choices to client, project manager, mentor and fellow proEP groups is key.

Learning outcomes

After successful completion the student is:

Able to perform a top down domain analysis of a distributed system which is fairly complex.

Able to make a justified choice for platforms and technologies necessary to be able to implement a system from point 1.

Implement a distributed application based on points above.

Able to work in a group with extensive responsibilities.

Entry requirements

Successful completion of:

Year 3 of the software engineering studies.

Internship.

Examination and grading

Examination

The final mark will consist of the group mark (50%) and the individual mark (50%).

The group mark comprises:

50% à products of the project; this is the straight average of the marks for:

- Project Plan
- Requirements Document
- Test Plan
- Design Document
- Software, User Manual, Technical Manual

The application of WCF (was it implemented correctly and in a logical way)

50% à project skills, teamwork and internal communication for the group as a whole.

The individual mark is derived from the group mark, using:

- the individual assessments of the tutor,
- the results of the last peer assessment.

The average of all individual marks will be the same as the group mark.

The assessment is based on both individual and group assessment.

Tools allowed

online information

sharepoint documentation.

Resits and repairs

Resit is possible in the next semester.

Grading

The final mark will consist of the group mark (50%) and the individual mark (50%).

The group mark comprises:

50% à products of the project; this is the straight average of the marks for:

- Project Plan
- Requirements Document
- Test Plan
- Design Document
- Software, User Manual, Technical Manual

The application of WCF (was it implemented correctly and in a logical way)

50% à project skills, teamwork and internal communication for the group as a whole.

The individual mark is derived from the group mark, using:

- the individual assessments of the tutor,
- the results of the last peer assessment.

The average of all individual marks will be the same as the group mark.

Teaching methods

project

Resources

Reader on Canvas, internet, forums, etc.

2.35. Information about TH Trends & Hypes

Content

write a report about a self chosen subject.

The structure of the template report has to be used.

Steps that have to be taken:

1. Subject, main research-question and end-product
2. subquestions, activitities (+strategies), phasing of the project
3. report structure
4. concept report.
5. final report

prerequisit: Internship has to be finished with a sufficient grade!

Learning outcomes

Learn :

- To do research about a self chosen ICT subject.
- to use the DOT researchframework for this research.
- to write report about it.

This must be done individually.

Examination and grading

Examination

The end mark is the mark for the final report (+ research product).

For the content the following aspects are important:

- Structure
- Subject description in chapter 2 of the report.
- Clarity of reasoning
- Use of sources, quality of the information
- Originality of your topic
- Use of English
- plagiarism

For a good mark and for the quality of your English it is important that you use your own words, and not start with a text from internet and change a number of words. Plagiarism is considered as fraud. Also know: previously awarded work cannot be reused.

Tools allowed

no end exam

Resits and repairs

insufficient means:

- rewrite the report the next semester.

Sometimes a new subject has to be choosen.

Grading

Grading for the wk1-, wk2-, wk3- and wk5-assignments are G (good), V (sufficient), or O (insufficient).

The end mark is the mark for the final report (+ research product).

Teaching methods

3 lectures.

individual assignments:

- handing in a subjectdescription (concept and final.
- handing in the structure of the report (+ introduction chapters)
- handing in the report (concept and final)

The teacher will give feedback on the assignments

Resources

powerpoint presentations

template report

Dot research documentation

websites etc.

2.36. Information about PRC2 Programming in C++

Content

- The C++ programming language, with focus on:
  - Parameter mechanism
  - Constructors (including copy-constructors),
  - dynamic memory allocation, (use heap <-> stack)
  - operator overloading
  - inheritance, multiple inheritance
  - virtual functions, abstract classes
- Using C++ with Arduino

Learning outcomes

- To be able to handle pointers and references.
- To understand the difference between the stack and the heap as memory regions.
- To be able to explain in which memory region -stack or heap-, declared variables and created objects are placed.
- To explain what memory leaks are and to be able to avoid them.
- To explain in which situations a destructor must be built and to be able to implement it correctly (that is, avoiding memory leaks).
- To be able to apply polymorphism using the “virtual” keyword correctly.
- To be able to explain and to apply the different purposes of the keyword “const”.
- To be able to use input/output streams to build simple text-based user interfaces.
- To be able to explain the difference between shallow and deep copying of objects, and to build appropriate implementations of copy constructors and assignment operators.
- To be able to create dynamic data structures and tear them down appropriately (that is, avoiding memory leaks).
- To apply inheritance and multiple inheritance properly.
- To be able to build abstract methods and pure abstract classes (that is, interfaces).
- To be able to explain what recursion is.
- To be able to analyse and to improve the execution speed of a program.

Examination and grading

Examination

practical assessment

Tools allowed

no additional tools are allowed

Resits and repairs

one resit in the same block, otherwise : next semester

Grading

practical assessment

Teaching methods

lectures with theory and practical

Resources

books, tutorials, slides, exercises, practical assignments

2.37. Information about PRC1 Programming in C

Content

- The C programming language, with focus on:
  - pointers,
  - arrays,
  - data structures,
  - strings,
  - bit manipulations,
  - file handling
  - call-by-value parameter mechanism
  - usage of pointers in fucntions
- Unit-testing

Learning outcomes

unit tests

To improve the quality of your code, at the end of this course you will be able

to apply unit tests and you will be able to review if the tests have a good coverage.

Note: in your previous C# courses you have worked with the Microsoft unit test

framework. In this course you will work with the Unity framework.

pointers

In this course you will learn the theory of pointers and how you can apply them.

This is related to the items Reference types vs value types in C#.

arrays

In this course you will learn how you can use C-arrays in your program;

in relation with pointers.

structs

You will learn what structs are and how you can apply them in a program.

file handling

In this course you will learn how to write and read from files in C.

strings

In this course you will learn how to work with C-strings.

bit manipulation

In this course your will learn the various bit manipulation operators.

You will learn when and how to apply them in your program.

Examination and grading

Examination

practical assessment

Tools allowed

your practical assignments; no additional tools are allowed

Resits and repairs

one resit in the same block, otherwise : next semester

Grading

practical assessment

Teaching methods

lectures with theory and practical

Resources

slides, video's, exercises, tutorials

2.38. Information about Graduation Project

Content

During the last semester of the bachelor course Information & Communication Technology the student must show he can function at the level of a graduate by completing a graduation project. This project comprises 90 - 100 working days and is normally carried out in the business community.

In this brochure you will be informed about the preparation, the realization and the evaluation of the graduation project period

There are only 5 possible starting dates per year for your graduation: Four at the first or second week of every quarter, and a fifth possibility in week 5 of the spring semester (feb-july). Below, all start dates are indicated by A,B,C,D and E. Accompanying graduation proceedings dates are indicated by A, B, C, D and E. In some cases the proceedings can also take place 1 week sooner.

Failure to start or finish a graduation on the dates below will automatically mean the start or finish will be postponed to the next scheduled possibility. To be able to start in a quarter, the student needs to have an approved survey on Wednesday of week 8 of the previous quarter. (for instance: for starting in September, you will need to have an approved survey before Wednesday in week 8 of the April-July quarter)

Learning outcomes

- Student is able to apply all teached knowledge during the graduation project
- Student is capable of applying the DOT framework research model
- Student can critical assess the environment of the company and set frequirements and scope by interviewing business people
- Student has the capability to advice the company with recommendations
- Student is capable of working by him / her self
- Student takes an entrepeneural attitude
- Student is able to adapt to the business and codes of conduct and regulations in the company
- Student can finish his / her product / project within 90-100 days
- Student is able to present the results in a process way with enough underpinning of choices before a jury

Examination and grading

Examination

Graduation project > 5.5

Presentation before jury > 5.5

Process Report > 5.5

Tools allowed

Not applicable

Resits and repairs

Student can, after approval of the FHICT Examboard, repair either the presentation or the report but not the work. All parts must be 5.5 or higher. In case of repair a maximum mark of 6 is awarded.

Grading

Student is assessed at the company by the university tutor and the company tutor for the work contents and attitude at the company

Student is assessed and questioned during his / her defence presentation before the jury

Student is assessed on his / her process report

Teaching methods

University tutor visits the company and the student at least two times. One time between week 4-6 and one time to rehearse the final presentation and discuss the mark for the work at the company

Resources

all learned at Fontys ICT

3. Information about Specialisation ICT & Technology

3.1. Information about MATH1 Mathematics 1

Content

- sets; algebra of sets
- counting principles; relations; mathematical induction
- logic, propositional calculus, truth tables
- quantifiers: universal, existential, sum, product

Learning outcomes

After completing this module the student is able to:

- calculate algebra of sets
- apply counting principles
- depict relations as sets, and vise versa
- proof a proposition via mathematical induction
- execute propositional calculus
- determine the truth table of a proposition
- calculate the result of these quantifiers: universal, existential, sum, product

Examination and grading

Examination

written exam

Tools allowed

table of laws is attached to the exam papers

Resits and repairs

next semester

Grading

final mark is the mark of the exam

Teaching methods

lectures for theory and practical

Resources

reader + slides

3.2. Information about PCS1 Programming in C# 1

Content

Subjects:

- Console applications
- Primitive data types
- Expressions, assignment statement,
- type conversions, operators
- Input/output
- Selection statements (if, switch and conditional operator)
- Loop statements (while and for loops)
- Classes

Learning outcomes

The student will gather knowledge and competences about programming in an object-oriented way.

Examination and grading

Examination

Exam on paper.

Tools allowed

whatever he/she brings in on paper.

Resits and repairs

exams in the exam-weeks of the odd-numbered blocks

Grading

mark for exam is final mark

Teaching methods

theoretical lessons and practica

Resources

Book Visual C# 2012 How to Program, Paul Deitel and Harvey Deitel

3.3. Information about EDB1 Databases 1

Content

- Introduction into relational databases

- Structured Query Language (select)

- Normalisation

- MS Access

Learning outcomes

The student is familiar with relational databases

The student can apply:

- normalisation

- select-command for answering queries

The student can work with MS Access

Examination and grading

Examination

Written exam.

Tools allowed

None.

Resits and repairs

Next semester.

Grading

Final mark is the mark of the exam.

Teaching methods

Lectures for theory and practical.

Resources

Reader and slides and optionally a book.

3.4. Information about FIS1 Fundamentals for ICT Students 1

Content

In this course we will work on getting your knowledge level to that of a starting IT student by covering the following topics:

- Basic hardware understanding;
- Basic operating system understanding;
- Calculus covering: Binary, hexadecimal en negative numbers;
- Background information about programming;
- Soft skills.

Learning outcomes

- You can explain how the hardware parts of a computer work as a whole;
- You can explain how a CPU works and what it's usage is;
- You can explain how a GPU works and what it's usage is;
- You can explain the different types of memory and their usage;
- You can explain what an OSand what kind of characteristics it can have;
- You can sum up the tasks of an complex OS;
- You can explain how multitasking works in an OS;
- You can explain how an OS handles memory for a processes;
- You can explain why processes cannot use the same resource at the same time;
- You can explain how a processes can communicate with an I/O device;
- You can explain what syntax, semantics and paradigm is;
- You can sum up different types of programming languages;
- You can explain what the usages of different types of programming languages are;
- You can sum up and explain the four main programming paradigms;
- You can describe what a low and high level programming language is;
- You can explain the differences between compiled, interpreted and JIT;
- You can describe the stages of compiled, interpreted, JIT compiled source code to running the application;
- You can explain what the usage advantages of an IDE are;
- You can do conversions between decimal, binary and hexadecimal;
- You can do addition and multiplication in binary;
- You can convert decimal to signed binary.

Examination and grading

Examination

The final grade for FIS1 will be based on the practical assignments and the written exam.

Tools allowed

No additional material is allowed during the written exam (no workbook, no calculator, no mobile phones, no other means are allowed).

Resits and repairs

If the final grade is not sufficient and you want to do a resit you need to redo both the practical assignment as written exam.

Grading

Description	Max. grade	Weight	
Practical assignments (1)		**100**	**30%**
	Group project	*100*	
	Week 5		
Written exam (2)		**100**	**70%**

1: The practical grade only counts if you manage to pass the practical of week 5 with a sufficient. If you did not get a sufficient for week 5 the practical assignments grade will be a 0. Even if you manage to get 100 points for the group project!

2: The written exam will be about the theory given in week 1 up until and including week 6. You can use the slides and your own notes to learn. Keep in mind that the slides have notes which gives an explanation about that slide. The exam will be 90 minutes long.

If the calculated final grade is not sufficient and you need to do a resit you need to redo both the practical assignment as written exam.

NOTE:

- The practical assignments must be completed on time
- You do **not** get any bonus points for the written exam

Teaching methods

In this course you will have a lecture part and a practical part. The lecture will span from week 1 up until and including week 5. As for the practical you're divided in to groups to work on a project. This project will be from week 1 up until and including week 4. As for week 5 and 6 you will work on individual assignments. Week 7 will be the week for recap of the lectures and a practice exam.

Handing in the practical will be done via email to your lecturer.

Course Planning for Students

Week	Lecture	Practical	Deadline deliverables to Tutor
Week 1	Intro module Intro basic hardware CPU, GPU & motherboard	Intro project Form groups and pick subject	Project group and topic
Week 2	Binary & hexadecimal	Exercises for binary and hexadecimal Work on project	Project report and presentation

Week 3	Memory & negative numbers	Project presentation	
Week 4	OS	Project presentation	
Week 5	Programs	Group exercise	
Week 6	Practical	Practical	Group exercise
Week 7	Theory recap	Practice exam	

Resources

- Lecture notes and ppt's
- Internet

3.5. Information about Dutch Introduction Dutch culture and language

Content

This module serves as introduction to living in the Netherlands. What is typically Dutch? In particular, the specific topics that come to discussion in this module are (in arbitrary order):

- Bikes
- Lichtjesroute/glow
- food
- culture shock
- ice skating
- Carnaval
- Orange (Dutch national color)
- Discounts
- Sinterklaas
- Dutch language

Learning outcomes

The student is able to:

- Communicate in Dutch in the following scenarios:
  - Purchasing products in a supermarket.
  - Greeting someone.
- Indentify and describe typical aspects of Dutch culture (See topics).
- Collaborate with other students to produce the final deliverable (video).

Examination and grading

Examination

Dutch finalizes with a presentation of the Video mentioned in the learning goals.

Tools allowed

There is no exam.

Resits and repairs

If a student is more than one lesson absent, or the video of the group is insufficient, the student is required to repeat the module.

Grading

The group will receive a sufficient if he/she satisfies the following requirements:

1 Attend at least six out of seven lessons (participating actively)
2 The spoken Dutch is understandable

The group will receive a good if he/she satisfies the following additional requirements:

1 The spoken Dutch is spoken fluently
2 The video is coherent

Teaching methods

Dutch is taught in a classical approach.

The final deliverable is made in groups.

Resources

Powerpoint presentations.

Language lessons on internet.

3.6. Information about MATH2 Mathematics 2

Content

- Linear Algebra: vector form and equation for lines and planes, matrix calculation, normal vector, dot product, cross product
- Automata: alphabet, words, languages, regular expressions, state diagram; finite state machines, powerset construction

Learning outcomes

After this course, the student is able to

- (Linear Algebra:) calculate the vector form and equation for lines and planes, and apply matrix calculation, normal vector, dot product, cross product
- (Automata:) indicate alphabet, words, languages; determine the language of regular expressions; draw a state diagram of a language; apply the powerset construction of a non-deterministic state diagram

Examination and grading

Examination

written exam

Tools allowed

no additional tools are allowed

Resits and repairs

next semester

Grading

final mark is the grade of the written exam

Teaching methods

lectures with theory and practical

Resources

slides, reader and exercises

3.7. Information about PCS2 Programming in C# 2

Content

Subjects:

- Windows applications
- Enumeration
- Overloading
- Arrays
- Collections, lists

Learning outcomes

- Windows applications
- Enumeration
- Overloading
- Arrays
- Collections, lists

Examination and grading

Examination

Laptop exam.

Tools allowed

everything on paper and on their own laptop

Resits and repairs

exams in the exam-weeks of the even-numbered blocks

Grading

mark for exam is final mark

Teaching methods

theoretical lessons and practica

Resources

Book Visual C# 2012 How to Program, Paul Deitel and Harvey Deitel

3.8. Information about EDB2 Databases 2

Content

- Datamodelling withe the use of Entity Relationship Diagrams

- Relational model

- Data Definition Language

- Data Manipulation Language (insert, update, delete)

- Constraints

- Oracle SQL Developer Data Modeler

Learning outcomes

The student can apply datamodelling with the use of ERD.

The student can transform a datamodel to a relational model.

The student can implement a relational model with the use of DDL, including simple constraints.

The student can modify the content of a relational database with the use of DML.

The student can work with Oracle SQL Developer Data Modeler (or another comparable tool).

Examination and grading

Examination

Written exam.

Tools allowed

None.

Resits and repairs

Next semester.

Grading

Final mark is the mark of the exam.

Teaching methods

Lectures for theory and practical.

Resources

Reader and slides and optionally a book.

3.9. Information about FIS2 Fundamentals for ICT Students 2

Content

In this course we will continue to work on getting your knowledge level to that of a starting IT student by covering the following topics:

- Testing and debugging;
- DLL's;
- Version control systems.

Besides these topics we will also give an introduction to the streams ICT & Business, ICT & Technology and ICT & Software. By giving background information and practical exercises you should get a feeling of what these streams are about.

Learning outcomes

- Explain the difference between ICT & Business, ICT & Technology & ICT & Software;
- Explain how to gather requirement from a client;
- Visualize a business process in a diagram;
- Create a simple algorithm for a delivery robot;
- What .NET Framework is and how C# fits into it;
- Explain what a Visual Studio project is and what files it has;
- Make use of a DLL in its C# projects;
- Make use of namespaces in a C# project;
- Explain what a version control system is;
- Work with GIT/GIT repository;
- Explain the difference between debugging and testing;
- Explain what a syntax, runtime, logical and efficiency error is;
- Simple debug techniques he/she can apply;
- Apply advanced debugging techniques like break points & watches;
- Create a test plan & report.

Examination and grading

Examination

The final grade for FIS2 will be based on the practical assignments.

NOTE:

- The practical assignments must be handed in before the deadline. Missing a deadline means a grade of 0;
- If plagiarism is discovered you will get a grade of 0 (insufficient) for that deliverable.

Tools allowed

All additional aids are allwed for practical assignments.

Resits and repairs

Re-sit is possible in the next semester.

Grading

You need to have at least a sufficient for deliverables from week 2 & 3 before you get an end grade.

Description	Max. grade	Weight
Week 1 - ICT & Business deliverable	10	30%
Project	10	70%

Teaching methods

This course consists of two hours theory and four hours practical (with two unsupervised) per week. The focus of this course will be about the practical. The theory will be given in week 1 until week 5.

Handing in the practical will be done via email to your lecturer.

Constraints

During week 1 groups will be formed and instructions will be given. If you're not present during this week the teacher will appointed you to a group. If missed week 1 and aren't present in week 2 you're out of the course.

During week 4 you will get a go or no-go about whether you can continue with the course. If you get a no-go you're out of the course.

Besides that you're required to be present for the first class of a week and also the first hour of the second class of the week. You miss more than one theory and one practical class you're out of the course. You're always required to hand in the deliverables before the set deadlines.

Course Planning for Students

Week	Lecture	Practical	Deadline deliverables to Tutor
Week 1	Intro course Intro ICT & Business Requirements gathering Process diagrams	ICT & Business assignment	Before Week 3: a document with the results of the practical
Week 2	Intro ICT & Technology Version control systems - Part A	ICT & Technology assignment	In week 3: show the deliverable to the lecturer
Week 3	Intro ICT & Software DLL	ICT & Software assignment	In week 4: show the deliverable to the lecturer
Week 4	Version control systems - Part B	Start project	
Week 5	Testing & debugging	Work on project	
Week 6	-	Work on project	
Week 7	-	Finish project	Before Week 8: Deliverables for the project
Week 9	-	Project event	

Resources

- Lecture notes and ppt's
- Practical readers & tutorials provided on Sharepoint
- Internet
- Lego Mindstorms

3.10. Information about PM Project Management

Content

-

week	lecture	Practical	reader	Deliverables to Tutor	Deliverables to group B
week 1	"Project Statement"	○ make questions for the interview with the formal client for the given case ○ prepare the interview	- study chapters 1 and 2	Questions for the client	

		with the formal client			
week 2	"Project Phasing"	○ project progress meeting tutor (per group, interview with client) ○ make mindmap and project plan version 1	- study chapter 3	Project plan version 1 Mindmap	
week 3		○ project progress meeting tutor (per group, feedback on project plan) ○ make additional questions for the interview with the formal client for the given case		Questions for the client	
week 4	"Project management"	○ project progress meeting tutor (per group, interview with client) ○ make project plan version 2		Project plan version 2	Project plan version 2
week 5		○ make feedback document for received projectplan		Feedback document	Feedback document
week 6		○ project progress meeting with other group (per group, receive your feedback) ○ make project plan Final version		Project plan Final	
week 7					

Learning outcomes

- Understand the importance of project management
- Understand the different tools for managing a project
- Able to ask the right questions to gather the requirements
- Able to reproduce a PID in coming projects
- Understand the different roles in a project

The goal of this course is to learn to write a Project Plan. You need to be able to write a project plan because you will make project plans (1) during your study and (2) at your future career. For example, in the next semester you will have to make a project plan at the beginning of the Project P-Phase (ProP). In this course the focus will be on writing a project plan within your career as a student of Fontys.

Examination and grading

Examination

In order to successfully pass the course, you must fulfil all requirements listed below:

1

Be present at lectures and progress meetings in all weeks (this course lasts 6 weeks). If you cannot attend the lecture/meeting for valid reasons (e.g., sickness), you must notify the teacher via e-mail before the lecture/meeting.

2

All deliverables must be submitted before the deadline, which will be given by your teacher during the lectures.

3
Mini exam in week 8/9 should be passed sufficient >=5.5

4
The Feedback document is marked Sufficient or more by the tutor.

5
Both the exam and the project plan should be graded with at least 5.5 or higher to pass Project Management course.

If you satisfy all assessment requirements, then your end mark for the Project Management course will is derived from the points for the Project Plan. These points are distributed as follows:

Total points = 10 points
1. Project Statement = 5 points
a. Client = 0,1
b. Leader = 0,1
c. Current situation = 1
d. Problem description = 1
e. Project goal = 0,5
f. Deliverables and Non-deliverables = 1
g. Constraints = 0,3
h. Risks = 1
2. Project Phasing = 5 points
a. Visual overview of the phases, activities, timeline, critical path = 1,5
b. Clear described tasks per activities = 2
c. Time estimation per activity = 0,5
d. Milestones per phase matching the activities= 1
-2,0 points for bad layout (title page, font, headings, alignment, etc.)
-1,0 points for a missing Title Page

Final mark for the Project Management course is derived from the points for the Project Plan and the exam.

project plan points	PM mark
9.5 – 10	10
8.5 – 9.4	9
7.5 – 8.4	8
6.5 – 7.4	7

5.5 – 6.4	6
4.5 – 5.4	5
3.5 – 4.4	4
2.5 – 3.4	3
1.5 – 2.4	2
0.0 – 1.4	1

Tools allowed

Books and materials

Resits and repairs

In case Project plan is >=5.5 and exam <5, retake of the exam next semester.

In case Project plan <= 5.5 whole course must be retaken

In case more than 1 class absent, whole course must be retaken

Grading

Total points = 10 points
1. Project Statement = 5 points
a. Client = 0,1
b. Leader = 0,1
c. Current situation = 1
d. Problem description = 1
e. Project goal = 0,5
f. Deliverables and Non-deliverables = 1
g. Constraints = 0,3
h. Risks = 1
2. Project Phasing = 5 points
a. Visual overview of the phases, activities, timeline, critical path = 1,5
b. Clear described tasks per activities = 2
c. Time estimation per activity = 0,5
d. Milestones per phase matching the activities= 1
-2,0 points for bad layout (title page, font, headings, alignment, etc.)
-1,0 points for a missing Title Page

Teaching methods

7 classes of 2 hours

Homework every week

Resources

- Reader: Project Management, an introduction
- Material on Sharepoint

3.11. Information about ICTA ICT Assessment

Content

The ICT assessment is intended to test integrally whether the learning outcomes of the first semester have been achieved. The assessment only takes place if there is any doubt about the learning outcomes or not. -

Learning outcomes

The learning objectives are the same as the learning objectives as described in the other modules of semester 1

Examination and grading

Examination

Type 5 and 6 testing. Assignments (good / suff / insuff) and oral exam (1..10)

Tools allowed

no extra tools

Resits and repairs

next semester

Grading

Assignments and oral exam are graded by two assessors.

Teaching methods

No special education form concerns only the assessment

Resources

all the lesson material of the other semester 1 modules

3.12. Information about PO/PD1 Professional Orientation and Personal Development 1

Content

The topics for this module are:

1 company visits,
2 cultural awareness,
3 presentation skills
4 introduction research framework

Learning outcomes

This course has two goals: To orient on the actual ICT work field and to improve the communication skills.

Examination and grading

Examination

In order to successfully pass the course, you must fulfil all requirements listed below:

1 Be present at lectures and progress meetings in all weeks . If you cannot attend the lecture/meeting for valid reasons (e.g., sickness), you must notify the teacher via e-mail before the lecture/meeting. An additional assignment will be given in week 8.
2 All deliverables must be submitted before the deadline, which will be given by your teacher during the lectures.

Tools allowed

no end exam.

Resits and repairs

if an assignment is insufficient the student can improve this once during the course.

if a student did miss a meeting, an additional assignment will be given in week 8.

Grading

If the student satisfy all assessment requirements, then your end mark for the PO/PD course will be “Voldoende”, meaning Sufficient.

Teaching methods

POPD1 works with group assignments and individual assignments.

Resources

powerpoint presentations

websites

DOT research documentation

3.13. Information about PCS3 Programming in C# 3

Content

Subjects:

- Inheritance and
- polymorphism
- Interfaces
- Unified Modelling Language (UML) class diagrams
- Exceptions
- Files
- Serialization
- Databases

Learning outcomes

The student will gather knowledge and competences about programming in an object-oriented way.

Examination and grading

Examination

Laptop exam.

Tools allowed

Everything on paper and his/her own laptop

Resits and repairs

exams in the exam-weeks of the odd-numbered blocks

Grading

mark for exam is final mark

Teaching methods

theoretical lessons and practica

Resources

Book Visual C# 2012 How to Program, Paul Deitel and Harvey Deitel

3.14. Information about ES1 Embedded Systems 1

Content

Embedded systems are at the core of innovations and new products that have great impact on our lives and our society. It requires a multidisciplinary field of engineering combining informatics, electronics, computers and control engineering. This course is a prerequisite for other engineering disciplines, such as industrial automation, mechatronics, robotics, automotive and the Internet of Things.

This course is the introduction to the basic technical aspects of embedded systems. Basic concepts of electronics will be taught and a diversity of sensors and actuators will be applied with an embedded development board – the Arduino Uno board. Special attention goes to the programming of the Arduino, the characteristics of sensors and actuators, and guaranteeing the safety of the electronic circuits and the Arduino.

By practical experiments, measurements and applied math, the physical aspects of electronic circuits, sensors and actuators are evaluated and written in a technical report. This includes software solutions. On the basis of such a technical report, solutions can be evaluated and improvements can be recommended. For example, recommendations can reduce the energy consumption of the system or improvements can increase the performance of the embedded software.

The content of the course is:

- Set up the Arduino development tools to program, build and flash programs on the Arduino development board.
- Programming the Arduino in C using Arduino libraries.
- Basics of electronics, such as voltages, currents, resistors, voltage dividers and Ohm’s law.
- Connecting hardware to the Arduino, such as led, button, servo motor, temperature sensor and distance sensor.
- Creating a fanciful embedded system using the hardware in the course.

Prerequisites:

- Programming in a C-like programming language, such as C, C++, C# or Java.
- Mathematics and physics at a level of higher general secondary education.

Learning outcomes

At the end of the course the student can:

- Program in C on an embedded platform:
  - you can use a variety of C constructs, such as functions, sequences, decisions, selections, and repetitions.
  - You can use different data types, such as primitive types, arrays and structures.
  - You can use variables within the correct scope in the program.
  - You can define functions to structure the program, to avoid duplication, to reuse code and to make code understandable.
  - You can document the code such that it makes the code readable and maintainable.
  - You can design and implement an algorithm for a given problem in which digital IO, analog input, PWM and bi-directional serial communication is applied.
- Set up electronic circuits for digital and analog sensor input on the Arduino.
- Set up electronic circuits for digital and PWM output on the Arduino.
- Explain and control a servo motor using PWM output.
- Design and program a protocol for bi-directional serial communication with text messages between the Arduino and a desktop computer.
- Set up user requirements, functional requirements and non-functional requirements for an embedded system.
- Combine multiple sensors and actuators in one fanciful application and keeping the code readable, scalable and maintainable.
- Trace errors in circuits and program code.
- Write a technical report.

Examination and grading

Examination

written exam + practical

Tools allowed

simple calculator

Resits and repairs

next semester

Grading

The final mark is the weighted average grade of the written exam and the practicum.

Teaching methods

The forms of learning are divided in theory lessons, practicum lessons, exercises, and individual coaching.

Resources

Slides, assignments, tutorials, datasheets and exercises.

3.15. Information about MATH3 Mathematics 3

Content

- (Isomorphic) graphs; Adjacency matrix; Eulerian trail, Fleury’s algorithm

- rooted trees, centroid, center; Search methods (DFS, BFS); (Minimum) spanning trees
- Shortest path algorithms; (Binary) trees; Prefix code
- Huffmans algorithm; Networks (capacity, flow)

Learning outcomes

After this course, the student is able to:

- calculate the adjacency matrix of a graph
- determine an Eulerian trail vua Fleury's algorithm
- calculate the centroid and center
- apply various Search methods (DFS, BFS);
- determine a (Minimum) spanning tree
- apply a shortest path algorithm
- calculate an optimal encoding of a string via Huffmans algorithm
- determine the maximum flow of a networks via the Ford-Fulkerson algorithm

Examination and grading

Examination

written exam

Tools allowed

1 double sided A4 with handwritten personal notes

Resits and repairs

next semester

Grading

final mark is the grade of the written exam

Teaching methods

lectures with theory and practical

Resources

slides, reader and exercises

3.16. Information about PO/PD2 Professional Orientation and Personal Development 2

Content

The topics for this module are:

1 information skills,
2 writing skills
3 DOT research framework: triangulatie

Learning outcomes

This course has three goals:

Improvement of a research project by learning:

- how to find information and when is it valid and reliable
- how to improve writing skills
- how the use of triangulation during a research project

Examination and grading

Examination

In order to successfully pass the course, you must fulfil all requirements listed below:

1 Be present at lectures and progress meetings in all weeks . If you cannot attend the lecture/meeting for valid reasons (e.g., sickness), you must notify the teacher via e-mail before the lecture/meeting. The assignment has to be done anyway.
2 All deliverables must be submitted before the deadline, which will be given by your teacher during the lectures.
3 No plagiarm allowed. All personal written essays will be checkt by Euphorus: copy paste will be considered as fraude.

Tools allowed

no end exam.

Resits and repairs

If an assignment is insufficient, the student can improve this once during the course.

Grading

If the student satisfies all assessment requirements, then the end mark for the POPD2 course will be “Voldoende”, meaning Sufficient.

Teaching methods

POPD2 works with a group assignment for the research topic and individual assignments for the rest.

Resources

powerpoint presentations

websites

DOT research documentation

3.17. Information about PCS4 Programming in C# 4

Content

Subjects: Event mechanism Delegates Searching and sorting Linked list, stack and queue

Learning outcomes

The student will gather knowledge and competences about programming in an object-oriented way.

Examination and grading

Examination

Laptop exam

Tools allowed

everything on paper and on his/her own laptop

Resits and repairs

exams in the exam-weeks of the even-numbered blocks

Grading

mark for exam is final mark

Teaching methods

theoretical lessons and practica

Resources

Book Visual C# 2012 How to Program, Paul Deitel and Harvey Deitel

3.18. Information about ES2 Embedded Systems 2

Content

Embedded Systems 2 is the follow up of Embedded Systems 1 with more in-dept understanding of the basics of electronics, new electric components, embedded software design and guaranteeing safety.

The content of the course is:

- Basics of electronics, such as applying Ohm's law and Kirchhoff's laws, voltage dividers, diodes, relays, amplifiers and filters that enhance the hardware interface of the embedded system.
- Proving safety by calculating the voltages, currents, and power that must be kept within the specifications of the Arduino and electric components.
- Set up protocols to establish reliable bi-direction communication between subsystems.
- Designing and implementing a well-behaved software execution framework using state machines and non-blocking techniques.
- Applying analog sensors (for example LDR, NTC).
- Applying amplifiers (for example relay, transistor and FET)

Prerequisites:

- ES1

Learning outcomes

The learning objectives in Embedded Systems 1 are inherited in this course. The learning objectives in this course is broken-down into common learning objectives and specific learning objectives per module.

Common learning objectives

The learning objectives continues with a more in-depth knowledge with new software techniques and new electric components.

At the end of the course the student can:

- Investigate and enhance the hardware interface of an embedded computer with elementary electric components useful for example to control domotica and traffic lights.
- Read and understand the more advanced tutorials (on the internet) about the Arduino and other embedded systems.
- Can explain and advise about the safety of the Arduino connected to sensors and actuators.

Basic of electronics

At the end of the course the student can:

- Explain and analyze electric concepts of current, voltage and resistor using Ohm's law, Kirchhoff's laws.
- Apply and explain basic electric components, such as capacitor, coil, relay, transistors, FETs and LED drivers, using datasheets.
- Design and implement a resistor network, voltage divider and amplifiers as part of the hardware interface of the embedded system.
- Proof the correctness and safety of electric circuits with calculations and measurements.
- Apply logical reasoning and quick reading of electric circuits, without calculates.

Protocols

At the end of the course the student can:

- Specify and develop a communication protocol between micro-controllers (and a computer) using a serial communication link with the following properties:
  - State machine
  - Nonblocking
  - Easy scalable
  - Acknowledgement of messages (ACK/NACK)
  - Timeout
- Identify and classify commands, states and functions that outline the protocol.
- Design, implement and compose behaviors with state machines to complete a reliable communication protocol between systems.
- Document the communication protocol.

Analog sensors

At the end of the course the student can:

- Use analog sensors (for example: NTC and LDR) and explaine the following concepts:
  - Hysteresis
  - Span
  - Explain in your own words how a capacitor behaves in the following situations:
- Direct current source (DC)
  - Alternating current source (AC)
  - Filters (RC-filter)

Stepper motor

At the end of the course the student can:

- Explain the characteristics of a one-phase and two-phase stepper motor
- Explain the H-bridge to control a stepper motor
- Control the stepper motor with an H-bridge and demonstrate the application

Examination and grading

Examination

written exam + practical

Tools allowed

simple calculator

Resits and repairs

next semester

Grading

The final mark is the weighted average grade of the written exam and the practicum.

Teaching methods

The forms of learning are divided in theory lessons, practicum lessons, exercises and individual coaching.

Resources

Slides, assignments, tutorials, datasheets, and exercises.

3.19. Information about NETWST Networks for S and T

Content

Subjects:

- Network Layered Model, TCP/IP
- IP Protocol, IP Addressing, IP Subnets
- TCP/UDP protocols
- DHCP/NAT protocols
- DNS protocol
- HTTP protocol
- All above mentioned protocols will be practiced in assignments
- Network configuration in Linux environment

Learning outcomes

Learning Objectives:

- The student can identify different layers of the TCP/IP protocols.
- The student can design network drawing including IP addressing
- The student can use basic Linux networking commands to configure a small IP network
- The student can explain different sort of routes used in static IP routing
- The student can interconnect different nodes of a small network by analyzing and using different types of static routing
- The student can explain difference between static and dynamic IP addressing
- The student can explain the use and the phases of DHCP protocol
- The student can demonstrate the working of DHCP protocol in a networking scenario
- The student can explain the use of NAT protocol
- The student can demonstrate the working of NAT protocol in a networking scenario
- The student can configure basic DNS server in a Linux environment
- The student can name the basic types of DNS records
- The student can explain the basic working of the DNS protocol
- The student can name the basic advantages of the IPv6

Examination and grading

Examination

- all assignments have to be completed in Canvas
- written multiple choice exam

Tools allowed

- no extra materials are allowed on the exam

Resits and repairs

- in the next semester

Grading

The final mark is a composition of 0-70 points for the exam and 0-45 points for the practical assignments in Canvas.

The passing criteria for this course is at least 30 points for the exam.

Teaching methods

- lectures with theory and practical done and submitted in Canvas

Resources

Olivier Bonaventure: Computer Networking : Principles, Protocols and Practice Netkit Tool, Wireshark, VMWare

3.20. Information about PROP Project P-phase

Content

Project P-phase (ProP) is a second semester module in which the students work in a group. The group is tasked with the development of a software solution that will facilitate the management of an event.

Pre-requisites:

- Completed the modules with a sufficient: EDB1, EDB2, PCS1 and PCS2.

Learning outcomes

The student should be able to:

- Design a software solution and plan a project based on a case.
- Implement the final product according to the design and plan.
- Implement the applications, website and database as one coherent system.
- Justify decisions that are made in the project.
- Apply version control in GitLab to the project according to FIS2 guidelines.
- Communicate effectively with the client.
- Conduct meetings in orderly/organized fashion.
- Collaborate effectively with group members within a project.

Examination and grading

Examination

The process is assessed by the tutor and a client (second tutor). At the end of the project, the group presents their solution to the client and reflects on the project. After this presentation the tutor and client grade the group. The tutor and client could decide to apply individual grades to (part of) the group.

Tools allowed

Not applicable.

Resits and repairs

If a student failed ProP, he/she will have to restart it during the next semester. The student will have to participate in a new group.

Grading

The final grade ranges between 1.0 - 10.0.

Criteria can be found in the current version of the student workbook (appendix C: Assessment Sheet).

The grade is dependent on the quality of the process and deliverables of the project.

Teaching methods

The student will work in group formation on the project. Teachers tutor this process.

Resources

Students can find a workbook in the shape of a PDF file on sharepoint.

3.21. Information about PRC1 Programming in C

Content

- The C programming language, with focus on:
  - pointers,
  - arrays,
  - data structures,
  - strings,
  - bit manipulations,
  - file handling
  - call-by-value parameter mechanism
  - usage of pointers in fucntions
- Unit-testing

Learning outcomes

unit tests

To improve the quality of your code, at the end of this course you will be able

to apply unit tests and you will be able to review if the tests have a good coverage.

Note: in your previous C# courses you have worked with the Microsoft unit test

framework. In this course you will work with the Unity framework.

pointers

In this course you will learn the theory of pointers and how you can apply them.

This is related to the items Reference types vs value types in C#.

arrays

In this course you will learn how you can use C-arrays in your program;

in relation with pointers.

structs

You will learn what structs are and how you can apply them in a program.

file handling

In this course you will learn how to write and read from files in C.

strings

In this course you will learn how to work with C-strings.

bit manipulation

In this course your will learn the various bit manipulation operators.

You will learn when and how to apply them in your program.

Examination and grading

Examination

practical assessment

Tools allowed

your practical assignments; no additional tools are allowed

Resits and repairs

one resit in the same block, otherwise : next semester

Grading

practical assessment

Teaching methods

lectures with theory and practical

Resources

slides, video's, exercises, tutorials

3.22. Information about OOD1 Object Oriented Development 1

Content

Subjects:

- UML Use-Cases
- UML Class diagrams
- UML sequence diagram
- Painting in C#

- Unit testing

Learning outcomes

The student will gather knowledge and competences about UML and implementing in an object-oriented way.

Examination and grading

Examination

written exam + practical

Tools allowed

nothing except your own brains

Resits and repairs

exams in the exam-weeks of the odd-numbered blocks

Grading

total of score for practica and exam, divided by 10 and rounded to the nearest integer

Teaching methods

theoretical lessons and practica

Resources

- lecture notes and ppt's
- Book UML Distilled, A brief guide to the standard Object Modelling Language, Third Edition, Martin Fowler,

3.23. Information about ES3 Embedded Systems 3

Content

A robot car, named RP6, is subject of study. It is a small autonomous electric car on batteries with two microprocessors and lots of electronics. The two microprocessors communicate with each other in order to control all sensors and actuators. The focus in this course is on programming the autonomous robot car, using the sensor characteristics and enabling the car to drive smoothly along a wall.

The content of the course is:

- Reading the datasheets and lab manual of the RP6 and sensors.
- Applying two-wire interface I2C to make one microprocessor responsible for moving and collision detection, and the other microprocessor responsible for supervisory control.
- Investigating the characteristics of the distance sensors on the RP6.
- Applying filter algorithms to make the RP6 insensitive to undesired disturbances.
- Applying a PID controller to make the RP6 autonomously and smoothly drive along a wall.

Prerequisites:

- ES1
- ES2

Learning outcomes

There is a breakdown between common learning objectives and specific learning objectives per module. The other learning objectives are found in the module.

Common learning objectives

At the end of the course the student can:

- Set up a prepared Linux-based development environment (virtual image) to build, program, and flash the RP6.
- Programming in C/C++ using the RP6 software development kit.
- Reading the manuals of the RP6 and collecting relevant information that is required for driving the RP6 along the wall.
- Writing a technical report of a more complex device, such as the RP6, than in the previous Embedded Systems courses.

Embedded programming

At the end of the course the student can:

- Apply Bit manipulation
    - Being able to manipulate bytes by reading and changing individual bits.
    - Configuring and reading processor registers on the basis of digital and analog sensors using GPIO and ADC.
- Use event handling techniques to enable the car to react on stimuli.
- Can implement I2C communication.
- Can implement a PID controller.
- Set up a real-time execution framework,

Two-wire-interface

At the end of the course the student can:

- Apply the following communications:
    - Communicating the two onboard microprocessors via I2C.
    - Communicating a microprocessor with an Arduino via I2C.
    - Communicating a microprocessor with an accelerometer device via I2C.

PID control

At the end of the course the student can:

- Apply and tune a PID controller for the RP6.
- Implement and test a PID-controller in software.

Sensor characteristics

At the end of the course the student can:

- Declare and recognize the following sensor characteristics:
    - Span
    - Accuracy
    - Sensitivity
    - Offset
    - Drift

Transfer functions

At the end of the course the student can:

- Study the distance sensor:
    - Determine the transfer function of the distance sensor characteristics.
    - Implement the transfer function to get the optimal performance on the RP6.

Algorithms interpolation

At the end of the course the student can:

- Implement the following algorithm on an embedded system:
    - Linear interpolation

Algorithms moving average

At the end of the course the student can:

- Study the moving average filter
    - Apply and analyze the moving average filter and get the optimal performance on the RP6.

- Discover the different behaviors of moving average for different lengths.

Examination and grading

Examination

written exam + practical

Tools allowed

simple calculator

Resits and repairs

next semester

Grading

The final mark is the weighted average grade of the written exam and the practicum.

Teaching methods

The forms of learning are divided in theory lessons, practicum lessons and individual coaching.

Resources

Slides, assignments, tutorials, datasheets, and lab manuals.

3.24. Information about ES4 Embedded Systems 4

Content

This is a course on microcontrollers used in embedded systems. Using a data sheet you will write low level code to directly control the microcontrollers hardware like input, output, timers and interrupts. You will also investigate the execution model and power consumption. You will perform measurements with a digital oscilloscope, logic analyzer and multimeter to verify the correct working of your code.

Learning outcomes

-

The student can derive the behavior of an embedded target from research using primary and / or validated sources such as datasheets and application notes from (chip) manufacturers and OEMs. The conclusions drawn from this will be backed up by POCs when necessary.

- The student is able to demonstrate and report his solutions to his stakeholders. The correctness of these solutions is supported by measurements and observations that follow from measurement setups with measuring equipment and test software as far as is needed for evidence.
- The student makes a informed choice for interrupts and / or signals and demonstrates a correct implementation of the interrupt service routines and signal handlers.
- The student can explain in his own words how the execution model including interrupts of an embedded target works.

Prerequisites

Before taking this course, the student must have experience with / have knowledge of:

- Basic use of a simple embedded system like Arduino.
- Basic programming in C using standard libraries.
- Basic binary, decimal and hexadecimal calculus.

Examination and grading

Examination

Practical assignments and a written exam

Tools allowed

For the practical assignments, all tools are allowed. When existing code is used this has to be mentioned explicitly, including a reference. In addition it has to be accompanied by research to the relevance, quality and working of that code.

For the written exam, only a simple calculator is allowed.

Resits and repairs

next semester

Grading

Final mark is weighted average of written exam and practical assignments.

During the course the student will get feedback for each assignment.

Teaching methods

Lectures with theory and practical demo's

Resources

Presentations + assignments

3.25. Information about DESeng Distributed Embedded Systems

Content

Contemporary products and systems often consist of multiple separate parts with each an embedded processor on board. Such an embedded processor board has its own memory and input and output peripherals. The parts of a system must cooperate in a coordinated way, such that to the user it seems that he or she interacting with only one thing. In order to create such coordinated behaviour, the parts of a system have to exchange data and control.

This course explicitly focuses on the communication between these parts and how this communication is formed to support a distributed application in a robust manner. The course uses CAN networking as an example embedded networking technique to show how the need for robust embedded networking leads to a specific protocol-stack specification and implementation.

Learning outcomes

==== Prerequisites ====

When entering this course, the student must:

* Be able to use Arduino/Genuino boards and the Arduino programming environment. * Be able to configure the Arduino programming environment to support different types of Arduino/Genuino boards. * Have an active understanding of the C programming language. * Have a general understanding of the working of processors and microcontrollers like the ones used for Arduino.

===== Learning objectives =====

After concluding this course the student should:

* Point-to-point: * Be able to //recognise//, //describe// and //explain// UART serial communication. * Be able to //recognise//, //describe// and //explain// UART character format. * Bus Network: * Be able to //recognise//, //describe// and //explain// the differences between point-to-point communication and multiple access bus networks. * Be able to //recognise// and //describe// different ways to control access to multiple access networks. * CAN Bus Communication: * Be able to //recognise//, //describe// and //explain// the physical layer in CAN bus networks. * Be able to //recognise//, //describe// and //explain// media access control in CAN bus networks. * Be able to //recognise//, //describe// and //explain// the datalink layer in CAN bus networks. * Be able to //recognise//, //describe// and //explain// the general use of CAN bus networks. * Common Principles: * Be able to //recognise// and //describe// distributed embedded systems and related terminology. * Be able to //recognise//, //describe// and //explain// mechanisms to transmit and receive bits of information. * Be able to //recognise//, //describe// and //explain// mechanisms for robust transmission and reception. * Be able to //recognise// and //describe// the OSI basic reference model, specifically the datalink layer and the physical layer.

Examination and grading

Examination

The student makes practical assignments, both individually as in small teams. The student will receive feedback and formative judgements that the student will use to improve their knowledge and abilities with respect to the stated learning objectives. This will lead to an overall judgement at conclusion of the course. Formative judgement means:

Informal:

* The teacher can provide both solicited and unsolicited intermediate feedback. * Students will help on another while executing practical assignments.

Formal:

* The teacher provides formal feedback on submitted practical assignments.

Tools allowed

While executing assignments all provided tools and resources are allowed. The student may also make use of previous own art.

Resits and repairs

The full period can be used by the student to build up a portfolio to prove that he attained the learning objectives of this course. This is a continuous process of making assignments and receiving feedback to improve on this. This means that re-examination and 'repairs' will also continuously take place. It means that after conclusion of the course no possibilities exist to re-examination or repair.

Grading

After conclusion of the period the student will receive a judgment. This judgement takes into account the following dimensions:

* Knowledge and insight, and * Applying this knowledge and insight, divided into: * Advice * Analysis * Design * Realization * Management

Teaching methods

In general the teacher will start with a lecture on a (new) subject, and after that the students will make assignments in order to practice what is tought. The teacher will be (part-time, i.e. regularly but not always) available for questions about assignments and for support at occurring problems.

Resources

Most of the material is available through presentations that students can find at Sharepoint. The course also uses existing communication standards, made available through Canvas. For example:

* CAN bus standard 2.0 * ISO-7498, OSI Basic Reference Model

Apart from this it is highly recommended to purchase the book:

* Voss, Wilfried: "A Comprehensible Guide to Controller Area Network", 2nd Edition, 2008, Copperhill Technologies Corporation.

3.26. Information about EL Embedded Linux

Content

This is an introductory course on Embedded Linux. You will create an embedded linux distribution and write programs for it. Such a multi-process program controls a USB device (like a Xpad controller). The embedded system is controlled by a host system via the TCP/IP protocol

Prerequisites:

PRC1, NETWST

Learning outcomes

After completing this module the student is able to:

- build an Embedded Linux toolchain and distribution with the help of BuildRoot
- investigate a program for the presence of: deadlock, lifelock, starvation, race condition and can explain their dangers
- implement and cross-compile a program for a microcontroller target system with the help of makefiles
- configure and use a host/target network
- program a distributed embedded linux application with the following characteristics:
    - controlling a USB device (e.g. Xpad controller) via LibUSB
    - multi-process, where communication is realized with shared memory and/or semaphores
    - multi-system, where communication is realized with TCP/IP

Examination and grading

Examination

written exam + practical

Tools allowed

no additional tools are allowed

Resits and repairs

next semester

Grading

final mark is weighted average of written exam and practical

Teaching methods

lectures with theory and practical

Resources

slides, tutorials, exercises

3.27. Information about CSA Client-Server Applications

Content

In this course you will learn how to make distributed Client-Service applications by using Windows Communication Framework (WCF).

Prerequisites:

Sufficient mark for modules PCS1, PCS2 and PCS3.

Learning outcomes

Learning outcomes of this course are:

- The student can use WCF operation contracts to connect client and service applications.
- The student can use programmatic and administrative configuration of service end points.
- The student can connect a client with a service via WFC proxy.
- The student can automatically generate a WCF proxy in the client application.
- The student can use WCF data contracts to transfer data between the client and the service.
- The student can make services with multiple operation contracts and end points.
- The student can make a service application that calls back the client application (two directions communication).
- The student can use publish-subscribe event-based communication between service and client.

Examination and grading

Examination

During the lessons you will get assignments for which you will get formative feedback (to which extent did you achieve the learning goals) and feed-forward (constructive guidance on how to improve) from the teacher.

One summative examination in which you individually apply the acquired skills in a new Client-Service case. The final mark is in the range 1 – 10 (1 – 5 is insufficient; 6 – 10 is sufficient).

Tools allowed

All additional aids stored localy on your laptop are allowed to use. It is not allowed to access material on the intenet or from other students.

Resits and repairs

Re-take is possible in the next semester.

Grading

Student gets a mark on the exam in the range 1 - 10.

Teaching methods

The course lasts 7 weeks. Each week there are 2 + 2 contact hours with the teacher. Each topic is shortly introduced by the teacher. For each topic a practical assignment is available, which helps students to practice and for whcih students can get feedback from the teacher.

Resources

- Slides and practical assignments on the SharePoint.
- Book “Programming WCF services” by Juval Löwy.

3.28. Information about IA Industrial Automation

Content

- Develop applications for PLC to control actuators and sensors
- Set up a production line
- Understanding of industrial automation by coupling several PLC-controlled modules

Learning outcomes

- implement a PLC controlled system
- validate a PLC program
- evaluate a function specification and translate it into the most suitable programming language
- Een functionele specificatie van een proces *ontleden* en *vertalen* naar de meest geschikte programmeertaal.
- recall, classify, select and implement PLC programming languages (IEC 6-1131: Ladder, FB, SFC, ST (of SCL) and IL) .
- *recognize and identify* “real world” situations (behavior of actuators and sensors) and translate them into a PLC program
- define, implement and validate Communication (Field bus, Profinet) *between two systems*.
- recognize, describem compare and apply the architecture of PLC systems
- use a PLC development environment (STEP 7 Siemens)
- recognize, compare and apply PLC standard functionalities (Timers, Counters, SET, RESET)
- categorize and use an OPC server
- create an overview of a Proces and use and illustrate it.

Examination and grading

Examination

written exam and practical

Tools allowed

no additional tools are allowed

Resits and repairs

next semester

Grading

final mark is weighted average of written exam and practical

Teaching methods

lectures with theory and practical

Resources

slides, workbook, tutorials, manuals

3.29. Information about MATH4 Mathematics 4

Content

Essential math in preparation of the course Feedback Control Systems: exponents + logarithms, trigonometry, derivatives + integrals, complex numbers

Learning outcomes

After completing this course, the student is able to:

- solve exponential and logarithmic equations
- explain sine, cosine and tangent with the unit circle
- convert degrees into radians and vise versa
- draw a sine graph based on a function's offset, amplitude and periodicity, and vise versa
- calculate the derivative and integral of polynomial, exponential and trigonomic functions
- given either a position-time, a velocity-time or an acceleration-time diagram, calculate the other two diagrams
- calculate the sum, subtraction, division and product of two complex numbers
- convert a complex number in cartesian coordinates into polar coordinates or euler notation and vise versa
- calculate the nth root of a complex equation and draw the results in a graph

Examination and grading

Examination

written exam

Tools allowed

simple calculator + table with formulas

Resits and repairs

next semester

Grading

grade of the exam

Teaching methods

study at home + practical

Resources

on internet: book, websites, videos, exercises

on intranet: slides, exercises

3.30. Information about PO/PD3 Professional Orientation and Personal Development 3

Content

The topics for this module are:

1 research-1: - SPA - systematic problem analysis.
2 research-2: main question -subquestions - Dot framework strategies and Methods applied to an internship project.
3 CV, how to contact a company
4 SWOT

5 finding a company
6 how to write a good projectplan

Prerequisites

- Knowledge about the DOT Research framework and the five strategies (FIELD, LIBRARY, WORKSHOP, LAB, SHOWROOM).
- Knowledge how to write a report.
- POPD1, POPD2 and PM : sufficient
- experience in a project.

Learning outcomes

Enhancing critical thinking skills, practicing techniques involving them and applying them in the context of internship and graduation.

Examination and grading

Examination

Every week there are deliverables in the form of an assignment. Also exercises done during class hours are considered deliverables.

Tools allowed

no end exam

Resits and repairs

One meeting can be missed, since one assignment can be repaired. Missing 2 meetings means failing the course.

Grading

The student will receive a score for each exercise and assignment.

Exercises will be scored done/not done. Assignments will be scored Insufficient/Sufficient/Good.

The end result is a combination of all exercises and assignments done.

All assignments need to scored sufficient or higher. If not: One assignment can be repaired in the week following the end of the course.

Not done exercises will lead to 0.5 of a point being subtracted of the final score of the assignments.

Assignments are graded on a linear scale going from 3 (all assignments insufficient), till 10 (all assignments good). All assignments sufficient will lead to a 6.

Teaching methods

individual and group assignments

Resources

powerpoint presentations

websites

youtube video's

DOT research documentation.

internship/graduation surveys

3.31. Information about FCS

Content

- principles of Feedback Control Systems
- open loop vs. closed loop, static vs. dynamic behavior of a system
- transfer function, reduction rules (product, sum, loop redux)
- P, PI, PID controllers for a 1-order process: simulation and analysis

Learning outcomes

The student is able to:

- describe and explain the terms transfer function, process, control system, controlled process
- simplify, calculate and evaluate 1st order systems and processes as well as 1st order approximations of systems and processes
- do measurements on processes and systems to empirically characterise the behaviour of said processes and systems
- Make and describe models o processes and systems that may be controlled or not in the s-domain
- Analyse 1st and 2nd order systems by calculation, evaluation, modelling and simulation
- Execute simulations of control systems with Matlab and Simulink
- Determine transfer functions of components, simple circuits, processes and systems including bode plots and phase diagrams.

The student can explain and reason about:

- Closed loop and open loop control systems
- Step-, constant value and ramp functions
- The response to op step-, constant value and ramp functions
- The behaviour and workings of a P-, PI- and PID-controller
- Electrical and physical transfer functions of components, processes and systems
- the relation between amplitude, frequency and phase of a periodic signal

Examination and grading

Examination

written exam

Tools allowed

nothing except pen and paper

Resits and repairs

next semester

Grading

Grade of written exam. A 60% score or higher is needed to pass the exam.

Teaching methods

Lectures with theory and instruction. Students are expected to active participate in the lectures.

Resources

Course material is provided via Canvas including slides, tutorials and practical exercises.

3.32. Information about ESPR Embedded Systems Project

Content

This course is on learning the C++ programming language and on applying C++ in an embedded systems project.

- The C++ programming language, with focus on:
  - Parameter mechanism
  - Constructors (including copy-constructors),
  - dynamic memory allocation, (use heap <-> stack)
  - operator overloading
  - inheritance, multiple inheritance
  - virtual functions, abstract classes
- Using C++ with Arduino
- Based on user requirements, set up an embedded system including functional requirements, design and test plan.
- UML: class diagram, state transition diagram, sequence diagrams.
- In C++, develop software for an Arduino coupled with a Centipede device and a Laundry Machine simulator.
- Set up unit tests with mocks and stubs.

Learning outcomes

- To be able to handle pointers and references.
- To understand the difference between the stack and the heap as memory regions.
- To be able to explain in which memory region -stack or heap-, declared variables and created objects are placed.
- To explain what memory leaks are and to be able to avoid them.
- To explain in which situations a destructor must be built and to be able to implement it correctly (that is, avoiding memory leaks).
- To be able to apply polymorphism using the “virtual” keyword correctly.
- To be able to explain and to apply the different purposes of the keyword “const”.
- To be able to use input/output streams to build simple text-based user interfaces.
- To be able to explain the difference between shallow and deep copying of objects, and to build appropriate implementations of copy constructors and assignment operators.
- To be able to create dynamic data structures and tear them down appropriately (that is, avoiding memory leaks).
- To apply inheritance and multiple inheritance properly.
- To be able to build abstract methods and pure abstract classes (that is, interfaces).
- To be able to explain what recursion is.
- To be able to analyse and to improve the execution speed of a program.
- To be able to create, based on requirements, a UML design for a simple system.
- To know and understand the notation and symbols that correspond to the following UML diagrams:
  - use cases,
  - use case diagrams,
  - class diagrams,
  - sequence diagrams,
  - state diagrams.
- To be able to use and apply the UML diagrams listed above to create an object-oriented design, following some structured design process.

- To be able to criticise the quality of a design based on various design principles (SOLID).
- To be able to implement a UML design using C++.
- To know and understand the difference between stubs and mocks, and their use in unit testing.
- To be able to use and apply unit testing with stubs and mocks.

Examination and grading

Examination

Grading of the project outcomes.

Tools allowed

Hardware and software tools needed to develop the project as described in the contents section above (hardware: Arduino, Centipede, Laundry Machine simulator; software: C++/Arduino development environment, unit testing tools).

Resits and repairs

Next semester.

Grading

Weighted average of the project outcomes.

Teaching methods

- Lectures with theory and practical exercises.
- Project with tutor meetings.

Resources

books, tutorials, articles, slides, exercises, practical assignments

3.33. Information about Internship

Content

During the four year bachelor course Information & Communication Technology you will work as a trainee in a company (profit or non-profit).

This internship comprises 90-100 working days (planned in the second semester of the third year). Take enough time to prepare yourself!

In this brochure you will be informed about the preparation, the realization and the evaluation of the internship project period.

Learning outcomes

- Student is able to apply all teached knowledge during the graduation project
- Student is capable of applying the DOT framework research model
- Student can critical assess the environment of the company and set frequirements and scope by interviewing business people
- Student has the capability to advice the company with recommendations
- Student is capable of working by him / her self
- Student takes an entrepeneural attitude
- Student is able to adapt to the business and codes of conduct and regulations in the company

Student can finish his / her product / project within 90-100 days

Examination and grading

Examination

internship project > 5.5 (40%)

Presentation before university tutor and company tutor > 5.5 (20%)

Process Report > 5.5 (40%)

Tools allowed

Not applicable

Resits and repairs

Student can, after approval of the FHICT Examboard, repair either the presentation or the report but not the work. All parts must be 5.5 or higher. In case of repair a maximum mark of 6 is awarded.

Grading

Student is assessed at the company by the university tutor and the company tutor for the work contents and attitude at the company

Student is assessed and questioned during his / her presentation before the university tutor and the company tutor

Student is assessed on his / her process report

Teaching methods

University tutor visits the company and the student at least two times. One time between week 4-6 and one time to rehearse the final presentation and discuss the marks

Resources

all learned at Fontys ICT

3.34. Information about PTT6

Content

The Prof Assignment is a research-based project on which you will work throughout the semester. In the Prof Assignment, you will apply the knowledge and skills you have acquired in the various courses, and you will have to actively search for new knowledge that you might need in order to finish your Prof Assignment. In addition, you will work on your professional development. You will receive regular feedback on both your professional development and behavior as well.

The Prof Assignment takes place in the context of the FHICT Innovation HUB, in which companies and the FHICT Lectorates collaborate to find innovative solutions to known and new problems by working on research assignments. The assignment for the Prof Assignment is based on one of these research assignments. The closing event of the Prof Assignment will be the "Innovation Insight" fair, at which all projects groups have to show their results.

The general objectives of the Prof Assignment are:

- Applying the research framework to come up with better and substantiated solutions to a problem.
- Stimulate to learn new things, both within and outside the scope of the Prof Assignment and the courses.
- Practicing real-life assignments of a kind that you will also encounter as a professional in the work field.

- Creating a complete image of you as a student and a future professional (both with regard to knowledge, skills and professional behavior).

This semester will focus on the professional development of software for reactive embedded systems. You will be given the opportunity to prove that you are able to professionally work on an embedded systems project in a team. In the Prof Assignment, you will go through the entire developing cycle of a product.

Scrum will be used to manage the development process. This means that you will regularly receive feedback from the client and your teachers. This feedback may pertain to o the product, the process and your professional behaviour. You are expected to use the feedback to implement improvements.

Learning outcomes

Professionally developing a product for a client in a team. In which you will have to show that:

- You know how to apply research skills in order to gain new knowledge necessary to realize the project and to justify choices that play a role in the design and realization of Prof Assignment products. And you show that you can use the research framework to come up with better and substantiated solutions to a problem.
- Together with your Prof Assignment group, you are able to develop a reactive embedded system that reacts to outside inputs. This input may come from sensors and other devices it can communicate with. You should incorporate subject matter knowledge from the courses (but not limited too) into the system.
- You are able to provide insight into the progress of the Prof Assignment and your own contribution to the Prof Assignment.
- You are able to reflect on feedback from the client, the tutor/semester coach, teachers and fellow students. In addition, you know how to use this feedback to improve your contribution to the Prof Assignment.
- You know how to apply Scrum to the Prof Assignment project.
- You can guarantee the quality of products by applying various testing, reviewing and design methods.
- You know how to use a version control system such as GIT or SVN to apply version management to code and documentation and the work on code and documentation in a team.

The abovementioned learning objectives will be used as input for the evaluation of the various assessment dimensions of the Prof Assignment, which are listed in paragraph 8.3.4. Because of its diverse character and the freedom you will get within the Prof Assignment project, it is also possible to add personal learning objectives in consultation with your Semester Coach.

Examination and grading

Examination

The Prof Assignment has periodic formative assessments that result in a formative indicator, which in turn will be part of the semester portfolio. The input for the formative assessment with regard to the Prof Assignment will come from the teachers of the various courses, the semester coach, fellow students and the external client

Informal:

- You can get feedback from your semester coach and course teachers at any time, either unsolicited or at your own request.

Formal:

- There will be peer reviews, in which students provide each other with feedback.
- There will also be audits, in which teachers will give feedback on products of the Prof Assignment.

- There will be progress interviews with your semester coach. The Personal Development Plan (PDP), and your reflection on the plan and its outcomes, will also be discussed during this interview.
- You will get feedback on your own functioning at regular interviews during the Scrum sprints.

The products that will be evaluated are: project plan, PDP, Scrum reports, audit reports, peer review reports, design documentation, research documentation, reflection reports, the actual product (or products) that you made for the client, the demonstration at the "Innovations Insight" fair at the end of the semester.

Tools allowed

- LPC3250, BeagleBone black or another embedded Linux platforms

Each project group will get a budget that they can use to order hardware. If specific hardware can be reused within a block it may be decided in consultation with the block owner to purchase the hardware for the entire block.

Resits and repairs

Students will have the entire semester to build a portfolio and they will receive regular feedback on the products they hand in. Students may also ask for feedback. This way students will get the chance to repair aspects of the Prof Assignment that they do not master yet throughout the course of the Prof Assignment project. Because of this, the Prof Assignment as a whole cannot be retaken.

Grading

At the end of the semester, students receive a formative indicator for their Prof Assignment, which they can incorporate into their semester portfolio. Because both the course subject matter and the students' professional development are part of the learning objectives, the student will be evaluated with regard to both aspects. In principle, the students of a project group will receive a collective formative grade with regard to the subject matter; however, this formative grade may deviate individually based on audits, peer reviews and professional conduct etc.

The assessment of the Prof Assignment will be based on the following assessment dimensions:

- Application of knowledge and insight
- Professional Development
    - Learning skills
    - Communication
    - Judgement

A formative grade will be given for each assessment dimension, using the USGO grading scheme. The formative grade of each dimension will be formed using a number of criteria. In the portfolio (paragraph: PDP) students will indicate for the various criteria how they intend to prove having mastered the criteria in question using performance indicators. At the end of the semester, the student will look back (reflection) and indicate how the criteria in question has been met.

The assessment dimension *Application of knowledge and insight* will weigh as half of the Formative indicator. The other three assessment dimensions will fall under the header *Professional Development* and together they will constitute the other half of the Formative indicator.

Teaching methods

The learning will take place by means of a project. Both the group and the individual student will get regular feedback from the semester coach, both in his role as the tutor of the Prof Assignment group and as the personal academic advisor who guides the individual student.

Resources

- The Prof Assignment manual
- The Scrum manual

3.35. Information about MDD6

Content

The software architecture of modern high-tech embedded systems generally is highly complex. Designing such complex systems usually requires a great bit of abstraction. The student therefore will have to learn how to design systems using software building blocks (interacting components).

When it comes to developing complex Embedded Systems, it is very important to detect any possible problems in an early stage. If not, the consequences of possible mistakes or incomplete specifications, which are inevitable, due to the complexity, will present themselves much later in the project. Solving these problems at that stage is a tiresome and expensive business.

Moreover, it is important to improve the productivity of embedded software engineers. The market is ever more demanding of embedded software. Manually writing and debugging software is a very labor-intensive process. The trick is not to work harder, but to work smarter.

Therefore, in the embedded software business demand arose for software engineers who know how to work with a Model-Driven Development approach. This approach puts the focus on a model of the software rather than the source code or the documentation. The code (and even documentation) will be generated from the model.

In the course the student will be acquainted with this Model-Driven Development (MDD) approach and learn how to design software using domain-specific modelling language.

Learning outcomes

The student will learn how to make embedded software, not by programming, but by modelling. You will learn to focus on the model of the software and to see the code as an artefact that you may generate from your model. You will learn to work with a software tool that supports this Model-Driven Development approach.

After having followed this course, you are able to:

- use any given modelling language and accompanying set of tools to make a model of an embedded system application, taking any given requirements as a starting point.
- validate the model and if possible verify it, insofar the modelling tool allows this,
- interface the model with a given embedded system platform and link it with existing software through hand-written code.
- generate and build code from the model and implement it on the particular platform.

Examination and grading

Examination

Assessment will be based on a series of assignments in which the student can prove having met the various learning objectives. The student will receive formative feedback for every assignment handed in in time. Every assignment the student hands in will be formatively evaluated using the USGO grading scheme.

Students may work together in couples to complete their assignments, but the formative assessment will be individual.

Tools allowed

The student will have access to a professional software-modelling tool and an embedded Linux system on which the software generated from the model should run. Students can also propose their own hardware platform. If they wish to do so, they need the permission from a teacher.

Resits and repairs

Throughout the semester the students will receive formative feedback on the assignments handed in, allowing them to continuously improve themselves.

There is a single repair opportunity for each assignment. A repair assignment will be graded with an S at best.

Grading

At the end of the semester, each student will receive a formative indication based on the evaluations for the assignments and the way in which the student has processed the feedback. Each assignment has its own weight in the formative indication.

Teaching methods

The classes will consist of a theoretic introduction and/or instruction by the teacher and a practical assignment in which the students will work on their modelling assignments.

The teacher will introduce the student to the concepts of Model Driven Development, the modelling language and the use of the modelling tool. The students will study the various sources of information they are given and practice modelling by carrying out a number of modelling assignments that grow increasingly in complexity:

In the first assignments, you will get familiar with the modelling tool and you will get the chance to train your design skills using abstract concepts. Subsequently you will learn how to generate code for an actual embedded system platform from an abstract model. Finally, you will do a final assignment to show that you are able to model and build a reasonably complex embedded application.

Teacher Independent learning is part of the teaching method. You will have to use methods of the research framework to gain knowledge and apply it when solving problems of the assignments.

Resources

The student will have access to the following learning materials:

- Digital learning materials (PowerPoint slides, assignment descriptions, project templates).
- A professional modelling tool.
- An embedded system.

Students moreover may refer to the following (non-obligatory) literature:

- A Practical Guide to SysML: The Systems Modelling Language, Authors: Sanford Friedenthal, Alan Moore, Rick Steiner, ISBN-10: 0123743796
- Systems Engineering with SysML/UML: Modelling, Analysis, Design, Author: Tim Weilkiens, ISBN-10: 0123742749

3.36. Information about ES6

Content

In the course ES3 you became familiar with an embedded Linux platform: how to set up such a platform, how to configure it and how to program it in user-space.

In the course ES6, you will get acquainted with the Linux kernel. Part of this course is writing device drivers (kernel modules) yourself and using different interfaces between kernel-space and user-space. Within the Linux kernel, you will use the C programming language.

Learning outcomes

Upon completing this course you will be able to:

- Develop Linux device drivers, which encompasses among others:
  - Using different interfaces between user space and kernel space.
  - Debugging your kernel module.
  - Understanding the difference between physical and virtual memory.
  - Writing a re-entrant code.
  - Using interrupts
  - Letting your kernel module sleep, awaking it again at the right moment.
- Write tidy, correct, expressive and well tested code.
- Measure correctness and speed of the implementation and select the right tools to do this (for instance an oscilloscope or a logic analyzer).
- Find out how stuff works and critically analyze your sources (e.g. data sheets) while being able to clearly document the results of your research.
- Develop a solution for an embedded product, where you decide what parts will be developed as kernel module and what part will be developed as user space application. You are able to convince your teacher that your solution works well by discussing your (system) design and by showing your test results.

Examination and grading

Examination

Every 2 or 3 weeks the students will get a new practical assignment on writing a device driver for a new piece of hardware of an embedded board. For these assignments, you will study how the hardware in question works and you will design a device driver, implement and test it.

You will hand in your implementation in Canvas, together with a report on your research, tests and self-reflection. You will have to demonstrate some of the assignments live during the practical sessions. The teacher will evaluate the assignment and provide the students with feedback, which they can use to improve their skills. When evaluating the assessments the teacher will take into account the extent to which feedback from the previous assignments was processed into the new assignment. The formative assessment of the assignment will be graded using the USGO grading scheme:

- G or O: If you continue this way, the formative indicator for the course in question will be Good or Outstanding.
- S: If you continue this way, the formative indicator for the course in question will be Satisfactory.
- U: If you continue this way, the formative indicator for the course in question will be Unsatisfactory.

Tools allowed

- An embedded board (ISSD)
- A Laptop (OSX / Windows / Linux) with the necessary compiling environment for the embedded board. There is a virtual Linux environment with everything you need on it available from your teacher.
- Various measuring devices (ISSD)

Resits and repairs

The retake is a substitutive, individual assignment. The scale and contents of the assignment will be attuned to the learning objectives the student has failed to achieve. Once the student has handed in this assignment, it will be determined whether or not the student has proven to master the subject matter.

Grading

The formative indicator for the course will be determined based on a combination of the learning objectives that the student attained. It should be noted explicitly that the formative indicator is not calculated by averaging the results of the assignments.

Students can positively influence their evaluation through:

- Quality (not quantity) of the research report
- Broader research than strictly necessary for the assignment
- Code quality:
    - clean code
    - well-structured code
    - code is to-the-point
    - avoiding code duplication
    - proper use of constants and defines (no magic numbers in the code)
    - as few comments as possible (code should be self-explanatory and should not have to rely on any comments)

Teaching methods

- Frontal (lectures, mostly 1 short lecture per assignment)
- Practical session with student-specific feedback
- Independent learning

Resources

- Sheets
- Documentation on various web sites
- Assignment documents
- Data sheets of the embedded board

3.37. Information about SD6

Content

During the SD6 course, you will get familiar with a number of techniques that are necessary to successfully develop software in a team. The techniques discussed are directly applicable in the Prof Assignment.

Learning outcomes

- You are able to describe the Scrum process in your own words and you are able to explain how this process is supporting the process of dealing with changes within a project aimed at creating value for the client.
- You are able to describe the role of Configuration Management and its subcomponents Change Management, Version Management and Release Management in managing changes within an Embedded Software project.
- You are able to set up procedures for the correct use of 'source control' and 'continuous integration' in a small-scale project, such as the Prof Assignment. In addition, you know how to apply these procedures.
- You are able to describe and use the various testing strategies.
- You are able to describe what a Design Pattern is and you can explain its purpose.
- You are able to characterize and implement a number of specific Design Patterns.
- You are able to elaborate on the strengths and weaknesses of a number of specific Design Patterns.

- You are able to implement an application in which you demonstrate the strength of a number of specific Design Patterns.

Examination and grading

Examination

The course will be assessed through assignments. These assignments will be made either individually or in couples of two students and they will be evaluated individually (with the exception of the assignments for the Prof Assignment groups). The couples will be announced at the start of the assignment or formed in agreement with the teacher.

The assignments will lead to products. These products will be rated with a formative indicator using the USGO grading scheme. The students will also receive feedback on the assignments. When evaluating the assignments the teacher will take into account the extent to which feedback was processed into the assignment.

The products to be evaluated are:

- A personal summary of the Scrum process, in which the roles, the activities, the products and objectives of Scrum are described.
- A presentation in which Configuration Management and its subcomponents Change Management, Version Management and Release management are explained and in which its relevance for problems to be solved in the Prof Assignment is explained.
- A fully installed and functional version management system that supports continuous integration and continuous testing, together with a process description for the user. This system and its procedures have to be applicable to a small-scale project, such as the Prof Assignment.
- A test plan in which the various testing strategies for the testing of an embedded system and its software are described. This testing plan has to be applied to a small-scale project, such as the Prof Assignment.
- The teacher will define the assignments for Design Patterns, but as the semester progresses students will increasingly create their own assignments. Assignments that students define themselves will have to be approved by the teacher.

Tools allowed

A PC with:

- A version management system of the student's choice.
- A continuous integration and testing environment of the student's choice.

Resits and repairs

Throughout the semester students will receive formative feedback on the assignments handed in, allowing them to continuously improve themselves.

There is a single repair opportunity for each assignment. To make use of a repair opportunity, students have to consult their teacher within a day after the assignment is graded, so that a repair plan can be made. A repair assignment will be graded with an (S) at best.

Grading

At the end of the semester each student will receive a formative indicator based on the evaluations of the assignments and how the student has processed the feedback.

Teaching methods

- Scrum workshop

- Lecture
- Instruction with feedback
- Independent learning

Resources

- Internet sources.

3.38. Information about Graduation Project

Content

During the last semester of the bachelor course Information & Communication Technology the student must show he can function at the level of a graduate by completing a graduation project. This project comprises 90 - 100 working days and is normally carried out in the business community.

In this brochure you will be informed about the preparation, the realization and the evaluation of the graduation project period

There are only 5 possible starting dates per year for your graduation: Four at the first or second week of every quarter, and a fifth possibility in week 5 of the spring semester (feb-july). Below, all start dates are indicated by A,B,C,D and E. Accompanying graduation proceedings dates are indicated by A, B, C, D and E. In some cases the proceedings can also take place 1 week sooner.

Failure to start or finish a graduation on the dates below will automatically mean the start or finish will be postponed to the next scheduled possibility. To be able to start in a quarter, the student needs to have an approved survey on Wednesday of week 8 of the previous quarter. (for instance: for starting in September, you will need to have an approved survey before Wednesday in week 8 of the April-July quarter)

Learning outcomes

- Student is able to apply all teached knowledge during the graduation project
- Student is capable of applying the DOT framework research model
- Student can critical assess the environment of the company and set frequirements and scope by interviewing business people
- Student has the capability to advice the company with recommendations
- Student is capable of working by him / her self
- Student takes an entrepeneural attitude
- Student is able to adapt to the business and codes of conduct and regulations in the company
- Student can finish his / her product / project within 90-100 days
- Student is able to present the results in a process way with enough underpinning of choices before a jury

Examination and grading

Examination

Graduation project > 5.5

Presentation before jury > 5.5

Process Report > 5.5

Tools allowed

Not applicable

Resits and repairs

Student can, after approval of the FHICT Examboard, repair either the presentation or the report but not the work. All parts must be 5.5 or higher. In case of repair a maximum mark of 6 is awarded.

Grading

Student is assessed at the company by the university tutor and the company tutor for the work contents and attitude at the company

Student is assessed and questioned during his / her defence presentation before the jury

Student is assessed on his / her process report

Teaching methods

University tutor visits the company and the student at least two times. One time between week 4-6 and one time to rehearse the final presentation and discuss the mark for the work at the company

Resources

all learned at Fontys ICT