Module Description Booklet

Bachelor Programs Information & Communication Technology

ICT & Software

ICT & Business

ICT & Technology

Version: September 2018

Joris Geurts, Frank Henning, Maja Pesic, Michèl van Eekhout

Table of contents:

1.Information about Specialisation ICT & Software Engineering

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

A) How is this topic examined?

written exam

B) Tools allowed

table of laws is attached to the exam papers

C) Resits and repairs

next semester

D) Determination of the grade

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

A) How is this topic examined?

Exam on paper.

B) Tools allowed

whatever he/she brings in on paper.

C) Resits and repairs

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

D) Determination of the grade

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

A) How is this topic examined?

Written exam.

B) Tools allowed

None.

C) Resits and repairs

Next semester.

D) Determination of the grade

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

A) How is this topic examined?

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

B) Tools allowed

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

C) 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.

D) Determination of the grade

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

A) How is this topic examined?

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

B) Tools allowed

There is no exam.

C) 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.

D) Determination of the grade

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

A) How is this topic examined?

written exam

B) Tools allowed

no additional tools are allowed

C) Resits and repairs

next semester

D) Determination of the grade

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 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

A) How is this topic examined?

Laptop exam.

B) Tools allowed

everything on paper and on their own laptop

C) Resits and repairs

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

D) Determination of the grade

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

1.8. 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

A) How is this topic examined?

Written exam.

B) Tools allowed

None.

C) Resits and repairs

Next semester.

D) Determination of the grade

Final mark is the mark of the exam.

Teaching methods

Lectures for theory and practical.

Resources

Reader and slides and optionally a book.

1.9. 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

A) How is this topic examined?

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.

B) Tools allowed

All additional aids are allwed for practical assignments.

C) Resits and repairs

Re-sit is possible in the next semester.

D) Determination of the grade

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

1.10. Information about PM Project Management

Content

•

week	lecture	Practical	reader	Deliverables to Tutor	Deliverables to group B
week 1	"Project Statement"	o make questions for the interview with the formal client for the given case o prepare the interview with the formal client	- study chapters 1 and 2	Questions for the client	
week 2	"Project Phasing"	o project progress meeting tutor (per group, interview with client) o make mindmap and project plan version 1	- study chapter 3	Project plan version 1 Mindmap	
week 3		o project progress meeting tutor (per group, feedback on project plan) o make additional questions for the interview with the formal client for the given case		Questions for the client	
week 4	"Project management"	o project progress meeting tutor (per group, interview with client) o make project plan version 2		Project plan version 2	Project plan version 2
week 5		o make feedback document for received projectplan		Feedback document	Feedback document
week 6		o project progress meeting with other group (per group, receive your feedback) o 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

A) How is this topic examined?

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

B) Tools allowed

Books and materials

C) 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

D) Determination of the grade

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

1.11. 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

A) How is this topic examined?

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

B) Tools allowed

no extra tools

C) Resits and repairs

next semester

D) Determination of the grade

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 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

A) How is this topic examined?

written exam + practical

B) Tools allowed

nothing except your own brains

C) Resits and repairs

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

D) Determination of the grade

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,

1.13. 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

A) How is this topic examined?

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.

B) Tools allowed

All additional aids are allowed.

C) Resits and repairs

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

D) Determination of the grade

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 - part a	Plan exploration 2 Work	feedback iteration 1 Planning iteration 2	Feedback iteration 1 (1 teacher, half class)

	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 Handin iteration 2 (2 days before first lesson next week)	Feedback exploration 2
6 - part a	Plan exploration 3 Work on exploration 3	Feedback iteration 2 Planning iteration 3 Work on iteration 3	Feedback iteration 2 (2 teachers, half class)
6 - part b	Plan exploration 3 Work on exploration 3	Work on iteration 3	Feedback iteration 2 (2 teachers, half class)
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

1.14. 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

A) How is this topic examined?

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

B) Tools allowed

N.A.

C) Resits and repairs

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

D) Determination of the grade

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/

1.15. 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 frond-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 multiuser 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

A) How is this topic examined?

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

B) Tools allowed

N.A.

C) 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.

D) Determination of the grade

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.

1.16. 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

A) How is this topic examined?

Laptop exam

B) Tools allowed

Laptop with SQL developer environment.

C) Resits and repairs

Next semester.

D) Determination of the grade

Mark for the exam is the final mark.

Teaching methods

Theory lessons + practical lessons.

Resources

Presentations and practical readers on Sharepoint

1.17. 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

A) How is this topic examined?

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

B) Tools allowed

None

C) Resits and repairs

Retake every semester. If passed for practical only exam retake

D) Determination of the grade

- 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)

1.18. 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

A) How is this topic examined?

- written exam (50%)

- case advisory report and 2 presentations (50%)

B) Tools allowed

Geen.

C) Resits and repairs

Volgende semester is er herkansing.

D) Determination of the grade

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 finaltest.
  - 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

1.19. 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

A) How is this topic examined?

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

E-project: a ready made and functional web shop

B) Tools allowed

none

C) Resits and repairs

retake every semester

D) Determination of the grade

Review of exam and case study. 2 Separate marks

Teaching methods

7 classes of 4 hours

Resources

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

1.20. 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.

o

Balance Sheet

o

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

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

Examination and grading

A) How is this topic examined?

- 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

B) Tools allowed

pen and calculator

C) Resits and repairs

Next semester retake

D) Determination of the grade

- 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

1 classes of 4 hour with theory, presentations, assignments

Resources

-

From bookboon

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

1.21. 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

A) How is this topic examined?

- 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

B) Tools allowed

pen and calculator

C) Resits and repairs

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

D) Determination of the grade

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

1.22. 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

A) How is this topic examined?

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

B) Tools allowed

N/A

C) Resits and repairs

In the next semester

D) Determination of the grade

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)

1.23. 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 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

A) How is this topic examined?

Activity

- weekly assignments(check the share point IPV assignment part), to be delivered in Canvas

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 **three assignments (week 2, 3 and 4)**, which students have to submit their corresponding assignment on canvas before the deadline, that is :

- These three assignments (week 2, 3 and 4) must be submitted on CANVAS on week3, 4 and 6 respectively.(**deadlines are strict! If you miss the deadline, you get 0 for that week**).
- For each of these three assignments (week 2, 3 and 4) students will get a grade:
- Assignments in weeks 1 and 5 are for students to practise.
- **The practical session of week3, 4 and 6 are mandatory.**
- Three assignments (week 2, 3 and 4) will be checked selectively during **the practical session** of week3, 4 and 6 respectively. Students should be able to demo the assignment that they submitted to the teacher and explain clearly what they have done. If not, your grade for that assignment will be 2.
- If **your .mlx file** in the assignment does not compile 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 2.

- It is allowed to have **at most one grade 5** for assignments in week 2, 3 and 4.
- The final IPV grade is the average grade for assignments in week 2, 3 and 4.

B) Tools allowed

NA

C) Resits and repairs

NA

D) Determination of the grade

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.

1.24. 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.

Examination and grading

A) How is this topic examined?

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.

B) Tools allowed

All additional aids are allowed.

C) Resits and repairs

Re-take is possible in the next semester.

D) Determination of the grade

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.

1.25. 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.

Examination and grading

A) How is this topic examined?

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.

B) Tools allowed

All additional aids are allowed.

C) Resits and repairs

Re-take is possible in the next semester.

D) Determination of the grade

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.

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

A) How is this topic examined?

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

B) Tools allowed

- Books
- Laptop
- Internet

C) Resits and repairs

Retake every semester of the written exam

D) Determination of the grade

- **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 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

A) How is this topic examined?

•Presentations + cases 40% (mark >=5)

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

•Total mark >=5.5 to pass!

B) 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.

C) Resits and repairs

Retake every semester

D) Determination of the grade

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.Information about Specialisation ICT & Business

2.1.   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

A) How is this topic examined?

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

B) Tools allowed

None

C) Resits and repairs

Retake every semester. If passed for practical only exam retake

D) Determination of the grade

- 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

2.2. 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

A) How is this topic examined?

Written exam + practical

B) Tools allowed

None.

C) Resits and repairs

Next semester.

D) Determination of the grade

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

2.3. 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

A) How is this topic examined?

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

B) Tools allowed

None

C) Resits and repairs

Retake every semester or resit of exam every semester

D) Determination of the grade

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

2.4. 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

A) How is this topic examined?

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

E-project: a ready made and functional web shop

B) Tools allowed

none

C) Resits and repairs

retake every semester

D) Determination of the grade

Review of exam and case study. 2 Separate marks

Teaching methods

7 classes of 4 hours

Resources

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

2.5. 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

A) How is this topic examined?

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

B) Tools allowed

- Books

- Laptop
- Internet

C) Resits and repairs

Retake every semester of the written exam

D) Determination of the grade

-

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

2.6. 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

A) How is this topic examined?

Written exam

B) Tools allowed

nothing except your own brains

C) Resits and repairs

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

D) Determination of the grade

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

3.Information about Specialisation ICT & Technology

3.1. 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

A) How is this topic examined?

practical assessment

B) Tools allowed

your practical assignments; no additional tools are allowed

C) Resits and repairs

one resit in the same block, otherwise : next semester

D) Determination of the grade

practical assessment

Teaching methods

lectures with theory and practical

Resources

slides, video's, exercises, tutorials

3.2. 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

A) How is this topic examined?

written exam + practical

B) Tools allowed

nothing except your own brains

C) Resits and repairs

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

D) Determination of the grade

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,

3.3. 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

A) How is this topic examined?

written exam + practical

B) Tools allowed

simple calculator

C) Resits and repairs

next semester

D) Determination of the grade

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.

3.4. 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, memory 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

After completing this module the student is able to:

- read a datasheet of a microcontroller
- configure and use the microcontrollers hardware like input, output, timers and interrupts
- minimize the power consumption of a microcontroller
- explain the execution model of a microcontroller
- explain the memory model of a microcontroller
- use a oscilloscope, logic analyzer and multimeter to verify the correct working of a microcontroller program

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

A) How is this topic examined?

Practical assignments and a written exam

B) Tools allowed

simple calculator

C) Resits and repairs

next semester

D) Determination of the grade

final mark is weighted average of written exam and practical

Teaching methods

Lectures with theory and practical demo’s

Resources

Presentations + assignments

3.5. 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

A) How is this topic examined?

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.

B) Tools allowed

While executing assignments all provided tools and resources are allowed. The student may also make use of previous own art.

C) 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.

D) Determination of the grade

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.

3.6. 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

A) How is this topic examined?

written exam + practical

B) Tools allowed

no additional tools are allowed

C) Resits and repairs

next semester

D) Determination of the grade

final mark is weighted average of written exam and practical

Teaching methods

lectures with theory and practical

Resources

slides, tutorials, exercises