Website design ideas

Must include:

* Front page
* Page for SDJ
* Page for RWD
* Page for DMA
* Page for SEP1
* Page for the Schedule (dummy schedule)
* Contact page with our faces;

PAGE FOR SDJ:

**Small description:**

Software Development with UML and Java

**Big description:**

The main purpose of the course is to provide students with the qualifications needed to understand the core object-oriented concepts and to implement smaller programs in Java from UML class diagrams.

**ECTS COUNT:** 10

**KNOWLEDGE:**

The student should be able to:  
  
• Identify the Java lexical structures: keywords, separators, operators, identifiers, literals and comments.  
  
• Explain details of UML class diagrams.  
  
• Identify selection and loop structures in UML activity diagrams.

**SKILLS:**

The student should be able to:

* Construct Java programs with proper choice of selection and loop structures.
* Create and use objects in Java.
* Implement classes in Java using the object oriented concepts: encapsulation, inheritance and polymorphism.
* Implement one-to-one relations and differentiate between association, aggregation and composition.
* Implement one-to-many relations using array structures and a simple collection class.
* Implement exception handling for different types of exceptions.
* Implement persistence in text and binary files.
* Construct simple event-based GUI applications.
* Construct Java source code documentations.

Interpret UML class diagrams, and construct corresponding Java code.

**COMPETENCES:**

 The student should be able to:  
  
• Exemplify and discuss basic object-oriented concepts, including encapsulation, relationships, inheritance and polymorphism  
  
• Implement small scale systems from UML class diagrams.

PAGE FOR RWD

Small description: Responsive Web Design

Big description: The purpose of this course is to introduce a set of theories and tools in order for students to obtain a proficient level of knowledge and gain a practical skillset for designing and developing responsive web sites for both PCs and mobile devices using basic web programming.

**ECTS:** 5

**Knowledge:**

Having completed this course, students will have the knowledge to:  
  
• Describe the different file formats used in web development and their purpose.  
  
• Reproduce webpage layouts using HTML5 and CSS3 when presented with images/screenshots of other websites.  
  
• Select appropriate attributes for HTML5 elements.  
  
• Explain the difference between responsive and non-responsive websites.  
  
• Test HTML5 files for errors using the W3C markup validator.  
  
• Account for the difference between the JavaScript and Java programming languages.

**Skills:**

Having completed this course, students will have the skills to:  
  
• Create web sites using Hyper Text Markup Language (HTML5).  
  
• Use simple and advanced CSS3 selectors and properties to style webpages.  
  
• Apply the Bootstrap grid framework to create responsive websites.  
  
• Utilize the Bootstrap classes to apply styling to responsive websites.  
  
• Implement JavaScript functions to add functionality to websites.  
  
• Use XMLHttpRequest to read content from an external source and integrate this content into a website.  
  
• Select HTML elements and apply jQuery animations to the selected elements to make websites interactive.

**Competences:**

Having completed this course, students will be able to:  
  
• Design and implement platform independent web applications.

Page for DMA

**Small description:**

Discrete Mathematics and Algorithms

**Big description:**

The aim of the course is to train students in the mathematical concepts and process of algorithmic thinking, allowing them to build simpler, more efficient solutions to real-world computational problems by building on the principles of mathematics.

**ECTS:** 5

**Knowledge:**

Upon completion of this course, students will be able to:  
  
• Describe fundamental concepts in number theory and modular arithmetic  
  
• Outline the basic principles of different sorting algorithms  
  
• Summarize key aspects of various data structures

**Skills:**

Upon completion of this course, students will be able to:  
  
• Give precise arguments for the correctness or incorrectness of an algorithm  
  
• Use key concepts of discrete mathematics for solving programming problems resourcefully  
  
• Analyze and compare the time and space usage of algorithms and data structures

**Competences:**

Upon completion of this course, students will be able to:  
  
• Adapt known algorithms and data structures to special cases of known problems or new problems  
  
• Design and implement small programs, using algorithms and data structures taught in the course.  
  
• Evaluate the performance of Java code with the objective of designing and implementing algorithms that optimise the code

PAGE FOR SEP

**Small description:**

Semester Project: Single User System

**Big description:**

To develop and document a single user system.

A basic introduction to study techniques and team-based project work.

**ECTS:** 10

**Knowledge:**

The student will use the knowledge acquired in SDJ1, RWD1 and DMA1.

Account for covered theories on group dynamics, team work and conflict resolution

Refer to covered theories on learning, motivation, feedback and study techniques

Identify and apply presentation techniques relevant for the target audience

Explain basic elements within Problem Based Learning

7Identify relevant problem statements and identify specific demands for a problem statement

Identify relevant project management methods, including planning, meeting management, risk assessment and quality assurance

**Skills:**

Explain the Waterfall method as a software development process

Apply use case modelling and draw activity diagrams

Draw a domain model

Construct class diagram(s)

Implement a software system using object-oriented programming

Integrate Java-generated files into a webpage using JavaScript

Perform testing in relation to the derived requirements

Describe how to use your system in a user guide

Explain and discuss the time and space usage of algorithms and data structures

Formulate and enforce a group contract together with the group

Establish and be part of a cooperation with the project group and the supervisor

Have a critical approach to sources, use references, apply proper reference management including comply with the rules for plagiarism

Communicate successfully in writing and orally to different target groups

Define a problem statement, describe different solutions and account for proposed solution

**Competences:**

Demonstrate the connection between the different stages in software development

Evaluate the performance of selected parts of their program in terms of time and space complexity using the Big O notation

Describe and reflect on the project group’s cooperation – including own effort – to define areas for improvement in future projects

Own learning process

Reflect on own ability to learn from different teaching and study activities including the project group’s work