POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

COURSE DESCRIPTION CARD - SYLLABUS

Course name
WWW and Scripting Languages [S1Teleinf1>WWWiJS]

Course

Field of study Teleinformatics	Year/Semester 2/4
Area of study (specialization) –	Profile of study general academic
Level of study first-cycle	Course offered in polish
Form of study full-time	Requirements compulsory

Number of hours

Lecture 15	Laboratory classes 30	Other (e.g. online) 0
Tutorials 0	Projects/seminars 0	

Number of credit points

3,00

Coordinators	Lecturers
dr inż. Robert Kotrys robert.kotrys@put.poznan.pl	

Prerequisites

he student starting this course should have basic knowledge of programming in High level languages. He should have the ability to program in High level languages. He should also understand the necessity of extending his competences. Moreover, in terms of social competence, a student must present such attitudes as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

The aim of the course is to familiarize the student with issues related to the design, development, maintenance and use of web applications. Developing skills of solving basic problems connected with data collecting and processing in the Internet application environment. Developing skills of acquiring knowledge about currently implemented solutions of digital information and communication systems.

Course-related learning outcomes

Knowledge:
Is able to use programming mechanisms and programming environments of object-oriented languages and available library software to develop web applications.

Can effectively use basic services of the Internet and design user interface for web applications using appropriate programming languages.
Is able to solve typical technical problems in the field of construction and operation of computer systems, operating systems, basics of database management and computer networks in order to effectively develop and operate web applications.
Is able to use basic computer techniques used in developing, testing, revising and documenting web application software.

Skills:
Knows basic data structures and algorithms used in web application programming languages and has a working knowledge of programming methodologies and techniques for such applications.
Has knowledge of object-oriented design and programming, the architecture of object-oriented programming systems, and basic object-oriented libraries in various programming languages, including those for writing web applications.
Has knowledge of effective use of services on the Internet, scripting programming languages, systems architecture, and tools and standards used in designing and building Internet applications.

Social competences:
Knows the limitations of his/her own knowledge and understands the necessity of its updating.
He/she is open to possibilities of continuous education and improvement of professional, personal and social competences, including web applications development.
He/she is aware of security threats to network systems and understands the need to use solutions that support data protection.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
In the field of laboratory, verification of the assumed educational effects is realized by:
- continuous assessment, at each class (oral answers),
- rewarding the growth of skills to use the known principles and methods,
- evaluation of the code to solve the task prepared partly during the classes, and partly after their completion. Tasks are evaluated in points from 0 to 10, passing requires at least 60% of the points.

In the scope of the lectures, verification of the assumed learning outcomes is realized by:
- assessment of knowledge and skills demonstrated in a written or oral test of a problematic nature, (3-5 questions from 50-60 questions available, the answer is evaluated in the scale of 0 to 10 points, credit requires at least 60% of the possible points.)

Programme content

Module activities are conducted in the form of lectures (15 hours) and laboratory classes (30 hours)
LECTURES
1. introduction, basic communication protocols, basic information about HTML5 and PHP scripting language
2. Elements of PHP, selected language constructs, examples.
3. Object-oriented programming in PHP.
4. CSS, web page styling.
5. JavaScript language.
6. JavaScript language - using JS on the example of jQuery component.
7. DOM Document Object Model. 8.
8. Elements of web application architecture on the example of Codeigniter design pattern.

LABORATORY EXERCISES
1. Getting familiar with the programming system available in the lab (Windows, editors, dedicated portal for publishing and testing the code)
2. creating a web page with structure compliant with the HTML5 language standard.
3. file operations managing data stored in files
4. Simple blog-type web application created using PHP language
5. Using CSS sheets to format data on the website.
6. Analyzing source code, session mechanism, participant management, participant authentication, and access control.

7. Introduction to object-oriented programming style in PHP.
8. Forum service, object-oriented programming style in PHP, separation of logic and presentation layers.
9. Getting acquainted with graphic functions library, creating graphics in PHP, file transfer methods in PHP
10. Learn about PDO middleware and how to use PDO objects.
11. learn SQL language and its applications, and the features of SQLite database. 12.
12. dynamic modification of a web page, JavaScript language, jQuery library
13 Practical introduction to data exchange between code executed in the browser "frontend" and code executed on the server "backend" using PHP, JavaScript and jQuery.
14. introducing the concept of application design pattern.
15. practical exercise in using the CodeIgniter design pattern as a framework for forum applications.
Module activities are conducted in the form of lectures (15 hours) and laboratory classes (30 hours)
LECTURES
1. introduction, basic communication protocols, basic information about HTML5 and PHP scripting language
2. Elements of PHP, selected language constructs, examples.
3. Object-oriented programming in PHP.
4. CSS, web page styling.
5. JavaScript language.
6. JavaScript language - using JS on the example of jQuery component.
7. DOM Document Object Model. 8.
8. Elements of web application architecture on the example of Codeigniter design pattern.
LABORATORY EXERCISES
1. Getting familiar with the programming system available in the lab (Windows, editors, dedicated portal for publishing and testing the code)
2. creating a web page with structure compliant with the HTML5 language standard.
3. file operations managing data stored in files
4. Simple blog-type web application created using PHP language
5. Using CSS sheets to format data on the website.
6. Analyzing source code, session mechanism, participant management, participant authentication, and access control.
7. Introduction to object-oriented programming style in PHP.
8. Forum service, object-oriented programming style in PHP, separation of logic and presentation layers.
9. Getting acquainted with graphic functions library, creating graphics in PHP, file transfer methods in PHP
10. Learn about PDO middleware and how to use PDO objects.
11. learn SQL language and its applications, and the features of SQLite database. 12.
12. dynamic modification of a web page, JavaScript language, jQuery library
13 Practical introduction to data exchange between code executed in the browser "frontend" and code executed on the server "backend" using PHP, JavaScript and jQuery.
14. introducing the concept of application design pattern.
15. practical exercise in using the CodeIgniter design pattern as a framework for forum applications.

Teaching methods

Lecture: multimedia presentation, supplemented with current examples and additional explanations on the blackboard
Laboratory exercises: practical solving of tasks which require implementation of selected elements of a web application. Source code analysis of the examples.

Bibliography

Basic:
1. Luke Welling, Laura Thomson; „Tworzenie stron WWW. Vademecum profesjonalisty”, Wydanie czwarte, Helion 2009.
2. Peter MacIntyre, Brian Danchilla, Mladen Gogala; „PHP. Zaawansowane programowanie”, Helion, 2012.
3. Kae Verens; „Projektowanie systemów CMS przy użyciu PHP i jQuery”, Helion, 2011.

4. Tom Negrino, Dori Smith; „Po prostu JavaScript”, wydanie VIII, Helion, 2012.

Additional:
1. "Zend Framework od podstaw. Wykorzystaj gotowe rozwiązania PHP do tworzenia zaawansowanych aplikacji internetowych", Autor: Włodzimierz Gajda
2. "Po prostu XML", Autor: Elizabeth Castro
3. "Projektowanie witryn internetowych dla urządzeń mobilnych", Autorzy: Gail Frederick, Rajesh Lal
4. "Magia interfejsu. Praktyczne metody projektowania aplikacji internetowych", Autor: Robert Hoekman jr
Breakdown of average student's workload
Hours ECTS
Total workload 86 3.0
Classes requiring direct contact with the teacher 45 2.0
Student’s own work (preparation for tests, preparation for laboratory 41 1.0
classes, literature studies)

Breakdown of average student's workload

	Hours	ECTS
Total workload	86	3,00
Classes requiring direct contact with the teacher	45	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	41	1,00