UNIVERSITY of NICOSIA

Course Syllabus

Course Code	Course Title	ECTS Credits
COMP-470	Internet Technologies	6
Prerequisites	**Department**	**Semester**
COMP-212, COMP-358	Computer Science	Fall, Spring
Type of Course	**Field**	**Language of Instruction**
Elective	Computer Science	English
Level of Course	**Lecturer(s)**	**Year of Study**
1st Cycle	Dr. Constandinos Mavromoustakis	4th
Mode of Delivery	**Work Placement**	**Corequisites**
Face-to-face	N/A	None

Course Objectives:

The main objectives of the course are to:

- Analyze Internet services, protocols, and World Wide Web architecture.
- Design client-server systems using socket programming and N-tier architecture.
- Develop web applications and evaluate usability/server configurations.
- Compare thin vs. thick client scripting paradigms.
- Evaluate HTTP protocol mechanics, including state management (RFC2965).
- Apply TCP/IP stack protocols and application interfaces.
- Design web architecture with URIs, representation management, and access control.
- Implement web caching using state-of-the-art techniques.
- Configure client caching, proxies (e.g., Squid Cache).
- Develop RESTful APIs with Java (Jersey v2.0+).
- Assess AI-powered CDNs for content delivery.
- Create web applications using HTML/XHTML/DHTML/XML.
- Implement client-side (JavaScript/VBScript) and server-side (Perl/ASP) scripting.
- Analyze socket programming models (Unix/Winsock/.NET).
- Research Semantic Web advancements and WWW technologies.

Learning Outcomes:

After completion of the course students are expected to be able to:

- Analyze communication protocols used in web technologies.
- Evaluate Internet infrastructure, underlying protocols, and services (e.g., World Wide Web).
- Critically assess TCP/IP architecture and application interfaces.
- Compare and contrast HTTP Protocol components (servers/clients, SSL, state management) against RFC standards.
- Investigate state-of-the-art research in Web caching, client-side caching control, and proxy technologies.
- Design and implement web applications using modern technologies (HTML, XHTML, DHTML, .NET).
- Develop N-tier web applications applying usability principles, Unicode, and methodology evaluation.
- Research caching techniques for streaming media in contemporary Internet architectures.
- Implement APIs for front-end/back-end mediation in application development.
- Construct RESTful APIs using modern tools and web standards.
- Research Semantic Web advancements and develop resource-constrained client applications.

Course Content:

1. Nature of the Internet. Internet Services and Protocols. World-Wide-WEB
2. Review of TCP/IP and application interface
3. HTTP Protocol. HTTP servers and clients, Hypertext Reference Model/RFC2965 - HTTP State Management Mechanism, HTML5 (cont'ed), CSS, CSS Fundamentals, Specifications, CSS Versions, visual representation of the CSS, Frameworks, Abstractions, etc.
4. Web Technologies and JavaScript, client-side scripting.
5. Web Sockets and Client/Server structures and client-side scripting using State management.
6. Sockets and Client/Server structures, N-tier architecture of the global internet. Servers and State management, Usability Principles, Methodologies & Evaluation, Unicode
7. TCP/IP stack and protocols (TCP/IP Tutorial, RFC 1180) and application interface
8. Architecture of the World Wide Web. Using a Uniform Resource Identifier (URI) to Access a Resource, Representation Management, URI persistence, Linking and access control

9. Web caching. Client site caching control. WEB Proxies. Web caching include additional configuration and administration of Squid Cache
10. Markup language using the XML & Web Technologies. Briefly cover the Web Programming: HTML, XHTML, Object Models, Styles, Dynamic content, DHTML
11. Programming the WEB: Client scripting. JavaScript, Jscript, VB Script
12. TCP/IP Protocol. Socket Programming paradigms using Java (Unix, Winsock, .NET)/System issues related to Internet programming and performance: protocols, client and servers, WWW interactivity, RESTful APIs examples, API as mediators for back-end and front-end application development, user semantic demands, Internet-based distributed systems
13. Peer-to-Peer Content Networks and Caching Techniques for Streaming Media/Acquire the knowledge about the RESTful APIs/A survey of current research on the reflective services

*Including Emulation/Simulation experiments for certain network scenarios using Core packages by Oracle, NS-3 and other integrated packages

Learning Activities and Teaching Methods:

Lectures, Lab Presentations, Lab Tutorials, Practical Exercises and Assignments.

Assessment Methods:

Tests/Quizzes, Design project, Homework, Project, Mid-Term, Final Exam.

Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Web Programming. Step by Step (2nd edition)	M. Stepp, J. Miller and V. Kirst	lulu.com	2018	ISBN-10:110557878X ISBN-13:978-1105578786
Interact with Web Standards. A Holistic	E. Anderson et al.	New Riders Pub	2021	ISBN-10:0321703529

Approach to Web Design				ISBN-13:978-0321703521
New Perspectives on HTML 5 and CSS: Comprehensive	Patrick M. Carey	Cengage, 8th Edition	2022	ISBN 9780357107140

Recommended Textbooks* / Readings:

Title	Author(s)	Publisher	Year	ISBN
Core Web Programming	M. Hall and L.Brown,	Prentice Hall	2017	0-13-089793-0
Web Technology: Theory and Practice	M. Srinivasan	Pearson	2018	ISBN-13: 9788131774199
Web programming and Internet technologies	P. Lingras and P. Scobey	Jones & Bartlett Learning; 2nd edition (Sept. 2016)	2016	ISBN-10: 9781284070682 ISBN-13:978-1284070682

*From the above recommended textbooks specific parts will be used during lecture sessions that will be specified by your lecturer.