

Course Number Change - Computer Science Curriculum

LSC Use Only No: <u>07-16u-</u> LSC Action-Date:	UWUCC USE Only No. <u>Info.-12/4/07</u> UWUCC Action-Date: <u>AP-10-30-07</u> Senate Action Date:
---	---

Curriculum Proposal Cover Sheet - University-Wide Undergraduate Curriculum Committee

Contact Person: David T. Smith	Email Address: dtsmith@iup.edu
Proposing Department/Unit: Computer Science	Phone: 7-4478

Check all appropriate lines and complete information as requested. Use a separate cover sheet for each course proposal and for each program proposal.

1. Course Proposals (check all that apply) <input type="checkbox"/> New Course <input type="checkbox"/> Course Prefix Change <input type="checkbox"/> Course Deletion <input type="checkbox"/> Course Revision <input checked="" type="checkbox"/> Course Number and/or Title Change <input type="checkbox"/> Catalog Description Change	
<u>Current Course prefix, number and full title</u> COSC 415 Internet Architecture and Programming	<u>Proposed course prefix, number and full title, if changing</u> COSC 365 Web Architecture and Application Development
2. Additional Course Designations: check if appropriate <input type="checkbox"/> This course is also proposed as a Liberal Studies Course. <input type="checkbox"/> Other: (e.g., Women's Studies, Pan-African) <input type="checkbox"/> This course is also proposed as an Honors College Course.	
3. Program Proposals <input type="checkbox"/> New Degree Program <input type="checkbox"/> Program Title Change <input type="checkbox"/> Other <input type="checkbox"/> New Minor Program <input type="checkbox"/> New Track <input type="checkbox"/> Catalog Description Change <input type="checkbox"/> Program Revision	
<u>Current program name</u>	<u>Proposed program name, if changing</u>
4. Approvals	
Dept Curriculum Committee Chair	<i>[Signature]</i> <u>10 Mar 06</u>
Department Chair	<i>[Signature]</i> <u>8/13/06</u>
Coll. Curriculum Committee Chair	<i>[Signature]</i> <u>05/17/07</u>
College Dean	<i>[Signature]</i> <u>9/26/07</u>
Director of Liberal Studies *	
Director of Honors College *	
Provost *	
Additional signatures as appropriate: (include title)	
UWUCC Co-Chairs	<i>[Signature]</i> <u>10/30/07</u>

* where applicable

Received

NOV 06 2007

Liberal Studies

Received

SEP 25 2007

Liberal Studies

Course Number Change - Computer Science Curriculum

Part II. Description of Curriculum Change

1. Catalog Description of course to be changed:

COSC 365 Web Architecture and Application Development

3c-01-3cr

Prerequisites: COSC 310 and COSC 341 or instructor's approval

Covers the fundamental architecture of Internet systems and the process of developing computer applications running on the Internet in general and on the World Wide Web in particular. Students gain a basic understanding of the TCP/IP protocols and the client/server technology. Methods, languages, and tools for developing distributed applications on the Internet are evaluated. Programming projects developing distributed applications, using a representative suite of development tools and languages, are an integral part of this course.

2. Current and Proposed Course Number, Title, and Prerequisites

Current (old) Course Number and Title:

COSC 415 Internet Architecture and Programming

Prerequisites: COSC 310 or instructor's approval

Corequisite: COSC 341

Proposed (new) Course Number and Title:

COSC 365 Web Architecture and Application Development

Prerequisites: COSC 310 and COSC 341 or instructor's approval

Other Changes:

There is no change to the content of the course.

3. Justification/Rationale

Change to Course Number

Web Architecture and Application Development provides competencies that are often needed for an Internship in Computer Science. Since an Internship in Computer Science is completed in either the last semester of the student's junior year or the first semester of the student's senior year, it is desirable that the course number assigned to this course be reflective of a course taken in the junior year and thus completed prior to an internship.

In the case that a student does not complete an Internship in Computer Science, the competencies covered by this class are often put into practice when a student takes COSC 320 Software Engineering Practice (this course is required when a student does not complete an internship for the applied track). It is desirable to have a course number assigned to this course be at the same or lower level than COSC 320.

Internet programming is a subject area that continues to expand within industry. Additional topics, such as XML and web services, continue to surface. Such topics are not currently covered in the Internet Architecture and Programming since the content of the course is already at the capacity of what can be reasonably covered in a semester. It is highly probable that a follow-on course to Internet Architecture and Programming will be needed in the near future. Such future course will be assigned a 400 level number. It is desirable to have a course number assigned to this course be at a lower level to allow for a follow-on course at the 400 level.

Change to Title

The title is being changed to better reflect the content of the course.

**Course Number Change - Computer Science Curriculum
Change to Prerequisite**

The previous catalog listing prerequisite/corequisite was inconsistent with the intent of curriculum committee. It was not intended that COSC 341 be a course that must be taken at the same time as this course. Instead COSC 341 was intended to be taken prior to COSC 415 or at the same time as COSC 415.

COSC 341 is now being listed as a prerequisite. Instructor consent will accommodate cases where 341 is taken at the same time.

Change to Course Outcomes

Minor revisions to course outcomes are made to reduce dependencies on specific technologies in the industry.

Change to Course Content

Revisions to course content are made reflect the developments in the field.

4. Letters or Acknowledgement

Attached is an email confirming that proposed number is available.

1. Syllabus of Record

I. Catalog Description

COSC365 Web Architecture and Application Development
Prerequisites: COSC 310 and COSC 341 or instructor's approval

3c-01-3sh

Covers the fundamental architecture of Internet based systems and development of computer applications running on the World-Wide-Web. Course presents the TCP/IP, UDP and HTTP protocols, HTML, and then progresses through different technologies to support dynamic Web applications. Main focus will be on one of i) J2EE architecture, ii) .NET architecture, or iii) other prevailing Internet architecture. Each of the technologies involved with the selected architecture will be critiqued with regard to ease of development, maintainability, and scalability. Issues such as state management, distributed processing, security, and internationalization will be discussed.

II. Course Outcomes

Upon successful completion of this course, the student will be able to:

- A. Explain and use basic building blocks for the Internet and Web including: sockets, datagrams, HTML/XHTML, HTTP, and Scripting (e.g. JavaScript, VBScript)
- B. List the major technologies of the selected Internet architecture and describe the purpose of each.
- C. Design and implement Web-based applications employing the technologies of the selected Internet architecture. Applications may include access and update of data in a database.
- D. Discuss problems and solutions related to Internet-based development such as security, privacy, state management, maintenance, scalability, and internationalization.
- E. Discuss the underlying framework for Internet-based software applications such as Web-based documentation retrieval systems, online transactions (such as banking, auctions, e-commerce, digital libraries, search engines, et al), group-based collaboration over the Internet, Web-based utilities (such as calendars, planners), Web-based entertainment, Web-based publishing, et al.
- F. Describe the evolution of existing Web technologies, as well as major future directions of new tools, techniques, applications, and paradigms for developing Web applications.

Course Number Change - Computer Science Curriculum

III. Detailed Course Outline

A. Fundamental Architecture of Internet-based Systems.	(3 hours)
1. Introduction to networking	
2. History of the Internet	
3. TCP/IP and sockets	
4. UDP and datagrams	
5. Distributed processing	
6. Remote Procedure Calls (RPC) and Remote Method Invocation	
B. Building Web Based Applications.	(10 hours)
1. HTTP protocol	
2. HTML/XHTML basics	
3. HTML/XHTML forms and controls	
4. HTML/XHTML tables	
5. Core technology of selected architecture to support dynamic we pages (e.g. Servlets)	
6. Server side scripting technology of selected architecture (e.g. JSP or ASP)	
7. Access and update of persistent data (e.g. a database)	
8. Three tier architecture	
9. State management strategies	
C. Client-side Programming	(3 hours)
1. JavaScript, VB Script, and/or other prevailing client side scripting language.	
2. The DOM representation of a web page	
3. Java Applets, Active X, and/or other prevailing client side plug in technology.	
4. Use of scripting for validations, enhanced interactivity, and/ or animations	
D. Developing Scaleable Enterprise Level Applications.	(9 hours)
1. Technologies to extend HTML tags (e.g. custom tags)	
2. Model 1 vs. Model 2 (i.e. Model View Controller MVC)	
3. Separation of business logic from presentation	
4. Security	
5. Internationalization	
6. Scalability	
7. Maintenance	
8. Review of sample applications (e.g. On-Line store application, E-Books, and Web-based Document Management)	
9. Introduction to a framework	
E. Team Project (or equivalent)	(12 hours)
1. Selection of an sample application	
2. Story board design of the pages composing the application	
3. Design and implementation of the application in a chosen framework	
F. Future directions on the Web and related topics.	(3 hours)
In class examinations	(2 hours)
Total	(42 hours)
Final	(2 hours)

Course Number Change - Computer Science Curriculum

IV. Evaluation Method

The final grade for the course will be determined as follows:

Exam 1 – course content to date	15 %
Exam 2 – rest of course content	15 %
Final – comprehensive	20 %
Various reinforcement projects	10 %
Contribution to the team developed Web application	30 %
Class participation	10 %

Grading Scale:

The following scale for the sum of the above scores will be used.

90 - 100%	A
80 - 89%	B
70 - 79%	C
60 - 69%	D
0 - 59%	F

V. Undergraduate Attendance Policy

Class attendance is regarded as being very important. Individual faculty may establish penalties for excessive number of unexcused absences. Excused absences will be allowed for illness, family emergencies, and involvement in university activities, such as sports. The penalties specified will meet university guidelines and be distributed to students with the course syllabus on the first day of class.

VI. Textbook

Java Enterprise in a Nutshell, Farley, Jim and Crawford, William, O'Reilly, Copyright 2005, ISBN 0-596-10142-2

Teach Yourself HTML and XHTML, Hayes, SAMS, 2002

VII. Special Resource Requirements

None.

VIII. Bibliography

Ablan, Professional Visual Basic 6 Web Programming, Wrox Press, 1999.

Active Server Page Corner, <http://www.kamath.com/>

Anderson, et al: Beginning Components for ASP, Wrox Press, 1999.

Bodoff, Green, Hasse, Jendrock, Pawlan, Stearns, The J2EE Tutorial, Addison-Wesley, 2002. Available on-line at:
<http://java.sun.com/j2ee/1.4/docs/tutorial/doc/index.html>

Chase, Active Server Pages 3.0 from Scratch, Que, 1999.

Client and Server Scripting in Web Pages, http://msdn.microsoft.com/library/techart/msdn_viscript.htm

Deitel, et al, Internet & World Wide Web – How to Program, Prentice Hall, 2000.

Eclipse, Eclipse.org, available at <http://www.eclipse.org>

Enterprise Java Beans and Java Server Pages

<http://developer.java.sun.com/developer/onlineTraining/J2EE/Intro/index.html>

Francis, et al: Beginning Active Server Pages 2.0, Wrox Press, Feb. 2000.

Friedrichs & Jubin: Java Thin-Client Programming for a Network Computing Environment, Prentice Hall, 1999.

Gold-Bernstein: Designing Enterprise Client/Server Systems, Prentice Hall, 1997.

Hunter & Crawford: JAVA Servlet Programming, O'Reilly & Associates, Inc., Sebastopol, CA, 1998.

Course Number Change - Computer Science Curriculum

Husted, Robert and J. J. Kuslich, *Server-Side JavaScript™: Developing Integrated Web Applications*,

<http://developer.netscape.com/tech/javascript/index.html?cp=dev01mtec>

Jakarta Struts, Apache Software Foundation, available at <http://jakarta.apache.org/struts>

Jakarta Struts Users Guide, Apache Software Foundation, available at <http://jakarta.apache.org/struts/userGuide/>

Joshi, et al: *The Comprehensive Guide to the JDBC SQL API*, Ventana Press, 1997.

Kaufman, et al: *Beginning ASP Databases*, Wrox Press Inc., 1999.

Kingsley-Hughes: *VBscript Programmer's Reference*, Wrox Press Inc., 1999.

Kurniawan, *Java for the Web with Servlets, JSP, and EJB*, New Riders, 2002.

Kurniawan, "Your First Struts Application", JAVAPro www.javapro.com, September 2002, available at

http://www.ftponline.com/javapro/2002_09/online/servletsjsp_bkurniawan_09_13_02

Kurniawan, "Manage Control Flow in Struts Apps", JAVAPro www.javapro.com, September 2002, available at

http://www.ftponline.com/javapro/2002_09/online/servletsjsp_bkurniawan_09_24_02

Kurniawan, "Move Onward and Upward With ActionForward", JAVAPro www.javapro.com, October 2002, available at

http://www.ftponline.com/javapro/2002_10/online/servletsjsp_bkurniawan_10_01_02

Kurniawan, "Build Classier Struts Apps With ActionMapping", JAVAPro www.javapro.com, October 2002, available at

http://www.ftponline.com/javapro/2002_10/online/servletsjsp_bkurniawan_10_08_02

Kurniawan, "Strut an ActionForm for All to See", JAVAPro www.javapro.com, October 2002, available at

http://www.ftponline.com/javapro/2002_10/online/servletsjsp_bkurniawan_10_15_02

Kurniawan, "Grace Under Errors", JAVAPro www.javapro.com, October 2002, available at

http://www.ftponline.com/javapro/2002_10/online/servletsjsp_bkurniawan_10_28_02

Laurie, et al (Editor): *Apache: The Definitive Guide*, O'Reilly & Associates, 1999.

Ligon: *Client/Server Communications Services: A Guide for the Applications Developer*, McGraw-Hill, 1997.

Moss: *Java Servlets With CDROM (2nd Ed.)*, McGraw-Hill Book Company, 1999.

Orfali & Harkey: *Client/Server Programming with Java and CORBA*, 2nd Edition, John Wiley & Sons, February 1998.

Patzner, Li, et al, *Professional Java Server Programming: with Servlets, Java Server Pages (JSP), XML, Enterprise*

JavaBeans (EJB), JNDI, CORBA, Jini and Javaspaces, Wrox Press, 1999.

Pekowsky, *JavaServer Pages*, Addison Wesley, 2000.

Rauch: *Open Systems Engineering: How to Plan and Develop Client/Server Systems*, John Wiley & Sons, 1996

Roman, *Mastering Enterprise JavaBeans and the Java 2 Platform, Enterprise Edition (with CD-ROM)*, John Wiley & Sons, 1999.

Seshadri, et al: *Enterprise Java Computing: Applications and Architectures*, Cambridge University Press, 1999.

Sessions: *Com and DCOM: Microsoft's Vision for Distributed Objects*, John Wiley & Sons, 1997

Siple: *The Complete Guide to Java Database Programming (Java Masters Series)*, McGraw Hill, 1997.

TechMetrix report on "Internet Development Tools & Application Servers",

http://www.techmetrix.com/products/prod_report1.shtml (Tools evaluated: WebObjects 4.0, Sapphire/Web 5.1, HAHTSite 3.1, Visual InterDev 6.0, NatWeb 2.0, Netscape App Server 2.1 Oracle App Server 4.0, Tango 3.5, PowerJ Ent. 2.5, Apptivity 3.0, Silverstream 2.0)

TechMetrix report on "Java Application Servers", http://www.techmetrix.com/products/prod_report2.shtml (In addition

to in-depth product comparisons, this report explains the real potential and positioning of the latest Java technologies: EJB, Java 2, Swing, ... ; Tools evaluated: Oracle, IBM, BEA, Inprise, Symantec, Gemstone)

Umar: *Object-Oriented Client/Server Internet Environments*, Prentice Hall, 1997.

Wang: *Java with Object-Oriented Programming and World Wide Web Applications*, ITP, 1999.

Course Number Change - Computer Science Curriculum

Old Syllabus of Record

I. Catalog Description

COSC 415 Internet Architecture and Programming

3 credits
3 lecture hours
0 lab hours
(3c-0l-3sh)

Prerequisites: COSC 304 and COSC 310 or instructor's approval.

Co-requisite: COSC 341

This course covers the fundamental architecture of Internet systems and the process of developing computer applications running on the Internet in general and on the world-wide-Web in particular. Students will first gain basic understanding of the TCP/IP protocols and the client/server technology. Methods, languages and tools for developing distributed applications on the Internet will be evaluated. Programming projects developing distributed applications, using a representative suite of development tools and languages, are integral part of this course.

II. Course Objectives

Upon successful completion of the course, the student will be able to

- A. Understand the architecture and operation of client/server applications in a multi-platform, multi-server networking environment.
- B. Design and implement simple Web-based applications employing distributed software components, including front-end clients, middleware, and back-end servers.
- C. Employ alternative tools and techniques for Internet-based system development, including
 1. Client side scripting such as JavaScript, VB Script, variations of HTMLs, and XML;
 2. Server side activation mechanisms such as Active Server Pages (ASP), Java Servlets, Java Server Pages (JSP), Server-side JavaScript, Server-side VBScript, and CGI (Common Gateway Interface).
- D. Design and implement database connectivity in a Web-based application.
- E. Gain an in-depth awareness of problems and solutions related to Internet-based development, such as security, privacy, and internationalization.
- F. Discuss various Internet-based software applications and their underlying framework, such as Web-based documentation retrieval systems, online transactions (such as banking, auctions, e-commerce, digital libraries, search engines, et al), group-based collaboration over the Internet, Web-based utilities (such as calendars, planners), Web-based entertainment, Web-based publishing, et al.
- G. Describe the evolution of existing tools such as component-based distributed standards, as well as major future directions of new tools, techniques, applications, and paradigms for developing distributed applications, such as distributed expert systems, mobile agents, and distributed databases.

III. Detailed Course Outline

1. Fundamental Architecture of Internet-based Systems. (3 hrs)
Introduction to networking, the Internet, and the World Wide Web. Brief introduction to the history and evolution of the Internet. The networking topologies and layers of networking software. The request and service relationship between two processes over the network. Introduction to the TCP/IP protocol suite and the HTTP protocol. Introduction to the concept of processes and inter-process communication (IPC). Discussion of TCP/IP Berkeley

Course Number Change - Computer Science Curriculum

sockets and its underlying mechanism of communication. Different types of sockets, including the stream sockets, datagram sockets, and sequenced packet sockets.

2. Client/Server Architecture, Sample Applications and Their Underlying Framework. (3 hrs)
Two-tier client/server model and examples. Client/Server as a special case of distributed computing. Database connectivity on the Web. Multi-tier client/server model and examples. Web-based documentation retrieval systems. Online transactions such as banking, auctions, e-commerce, digital libraries, search engines, et al. Group-based collaboration, planning, and development over the Internet. Web-based utility programs such as calendars, planners, et al. Web-based entertainment such as Web TV, music, movies, games, et al. Web-based publishing of books and multimedia information.
3. Client-side Programming Tools. (6 hrs)
Discussion of the existing client-side programming tools such as HTML, Dynamic HTML, VRML, XML, JavaScript, VB Script, and Java Applets. Study of sample programs and hands-on experience.
4. Web Servers. (3 hrs)
Discussion of major Web servers available in various platforms, such as Microsoft Personal Web Server and Internet Information Server (IIS), Netscape Enterprise Server, and the Apache server. Discussion of the role of a Web server in integrating the client and the server processes. Responsibilities of the Web administrator.
5. The Client/Server Enabling Technology.
 - a. Remote Procedure Call (RPC) (3 hrs)
Discussion of how the client process invokes a remotely located procedure (in the server process) and how the remote procedure executes and sends the response back to the client process. Discussion of sample programs. Analysis of the strengths and weaknesses of RPC.
 - b. Remote Data Access (RDA) and Web Database Connectivity (3 hrs)
Discussion of how the client programs and/or end-user tools issue database queries against remotely located databases. Discussion of database connectivity in various development tools (such as JDBC in Java, ADO in Visual Basic, and DBI in Perl/CGI). Discussion of legacy data access.
 - c. Message-Oriented Middleware (MOM) (3 hrs)
Discussion of the queued message processing (QMP) paradigm. Discussion of how MOM supports the QMP paradigm in a client/server environment. Analysis of the strengths and weaknesses of the QMP paradigm.
6. Server-side Programming Mechanisms and Tools. (6 hrs)
Discussion of server side programming mechanisms and tools, such as CGI (Common Gateway Interface), ASP (Active Server Pages), Java servlets, and JSP (Java Server Pages). Brief introduction to the Perl programming language. Study of sample programs and hands-on experience.
7. Design of Distributed Transaction Processing (DTP) (6 hrs)
Discussion of transaction processing concepts. The ACID properties of a transaction: atomicity, consistency, isolation, and durability. Discussion of serializability (concurrency control) and commit processing. Introduction to distributed transaction processing. Two-phase commit algorithm. Sample DTP applications.
8. Related Topics (3 hrs)
 - a. Distributed Objects and Component-Based Development
 - b. Application servers
 - c. Security issues: encryption, liability, audit, confidentiality, and integrity, et al.
 - d. Distributed database systems
 - e. Artificial intelligence: agents and distributed expert systems
 - f. Data aggregation using XML
 - g. Internationalization/Localization of information on the Internet

IV. Evaluation Methods

The final grade for the course will be determined as follows:

50% Examinations. Three in-class exams, each counting 9%; one final exam counting 23%.

40% Projects/Assignments. Four programming assignments, each counting 10%. These assignments are directly related to the objectives.

10% Quizzes. Five announced quizzes on the lecture material, each counting 2%.

Grading Scale: The standard grading scale will be used.

90%+ = A; 80-89% = B; 70-79% = C; 60-69% = D; below 60% = F.

Suggested Projects:

Project #1: Dynamic Web Page. (duration: 3 weeks)

Create a dynamic Web page using HTML and one of the client-side scripting languages such as JavaScript, Vbscript, or Java applets. Students would be required to create a dynamic Web page using one of the scripting languages.

Project #2: Client/Server Programming. (duration: 3 weeks)

Create a pair of client and server programs. Students would be required to write a basic client/server application. The client program is a dynamic Web page enhanced by a client-side scripting languages. The server program presents a form for online viewers to send their names, email addresses, and comments to the Web server. The server program would be implemented using one of the server-side programming methods, such as Java servlet, CGI, Java Server Page (JSP), or Active Server Page (ASP). The server program is in charge of receiving the data submitted by the client and appending the data into a file on the Web server.

Project #3: Web-based Database Connectivity. (duration: 3 weeks)

Create a pair of client and server programs involving the use of database server(s) on the Web. Students would be required to enhance their programs from project #2, or a sample solution to project #2 provided by the instructor, by saving multiple user data into a database, which is linked to the server program. In addition, the client program needs to be revised to allow the user to issue data retrieval/update queries. The server program needs to be revised such that it is capable of connecting to a database and submitting embedded SQL queries to the database server(s). The student would choose a connectivity module appropriate to their choice of server implementation mechanisms.

Project #4: Multi-tier Distributed Transaction Processing System (duration: 3 weeks)

Create a distributed transaction processing project. Students would be required to modify their programs from project #3, or a sample solution to project #3 provided by the instructor, by employing distributed transaction processing techniques. The server program would be able to process simultaneous read/write requests issued by multiple clients. Students would be required to design various data access patterns and validate the serializability property exhibited by the server program.

V. Required Textbook, Supplemental Books and Readings

Required

Umar, *Object-Oriented Client/Server Internet Environments*, Prentice Hall, 1997.

Flanagan, et al, *JAVA Enterprise in a nutshell: A Desktop Quick Reference*, O'Reilly, 1999.

Francis, et al, *Beginning Active Server Pages 2.0*, Wrox Press, 1998.

Course Number Change - Computer Science Curriculum

Supplemental

Deitel, et al, *Internet & World Wide Web – How to Program*, Prentice Hall, 2000.

Kaufman, et al: *Beginning ASP Databases*, Wrox Press Inc., 1999.

Sessions, Com and DCOM : *Microsoft's Vision for Distributed Objects*, John Wiley & Sons, 1997.

VII. Bibliography

Ablan, *Professional Visual Basic 6 Web Programming*, Wrox Press, 1999.

Anderson, et al: *Beginning Components for ASP*, Wrox Press, 1999.

Blackburn: *Professional Visual Interdev6 Programming*, Wrox Press, 1999.

Chase: *Active Server Pages 3.0 from Scratch*, Que, 1999.

Chase, *Active Server Pages 3.0 from Scratch*, Que, 1999.

Deitel & Deitel, *Java: How to program*, Prentice Hall, 1999

Deitel, et al, *Internet & World Wide Web – How to Program*, Prentice Hall, 2000.

Francis, et al: *Beginning Active Server Pages 2.0*, Wrox Press, Feb. 2000.

Friedrichs & Jubin: *Java Thin-Client Programming for a Network Computing Environment*, Prentice Hall, 1999.

Gold-Bernstein: *Designing Enterprise Client/Server Systems*, Prentice Hall, 1997.

Hunter & Crawford: *JAVA Servlet Programming*, O'Reilly & Associates, Inc., Sebastopol, CA, 1998.

Joshi, et al: *The Comprehensive Guide to the JDBC SQL API*, Ventana Press, 1997.

Kaufman, et al: *Beginning ASP Databases*, Wrox Press Inc., 1999.

Kingsley-Hughes: *VBScript Programmer's Reference*, Wrox Press Inc., 1999.

Laurie, et al (Editor): *Apache: The Definitive Guide*, O'Reilly & Associates, 1999.

Ligon: *Client/Server Communications Services: A Guide for the Applications Developer*, McGraw-Hill, 1997.

Minoli, et al: *Client/Server over ATM : Making Use of Broadband to Support Client/Server Applications*, Prentice Hall, 1997.

Moss: *Java Servlets With CDROM (2nd Ed.)*, McGraw-Hill Book Company, 1999.

Orfali & Harkey: *Client/Server Programming with Java and CORBA, 2nd Edition*, John Wiley & Sons, February 1998.

Ozsu & Valduriez: *Principles of Distributed Database Systems, 2nd Edition*, Prentice Hall, Englewood Cliffs, NJ, 1998.

Patzer, Li, et al, *Professional Java Server Programming: with Servlets, Java Server Pages (JSP), XML, Enterprise JavaBeans (EJB), JNDI, CORBA, Jini and Javaspaces*, Wrox Press, 1999.

Pekowsky, *JavaServer Pages*, Addison Wesley, 2000.

Rauch: *Open Systems Engineering: How to Plan and Develop Client/Server Systems*, John Wiley & Sons, 1996

Roman, *Mastering Enterprise JavaBeans and the Java 2 Platform, Enterprise Edition (with CD-ROM)*, John Wiley & Sons, 1999.

Seshadri, et al: *Enterprise Java Computing: Applications and Architectures*, Cambridge University Press, 1999.

Seshadri, et al, *Enterprise Java Computing: Applications and Architectures*, Cambridge University Press, 1999.

Sessions: *Com and DCOM : Microsoft's Vision for Distributed Objects*, John Wiley & Sons, 1997

Siple: *The Complete Guide to Java Database Programming (Java Masters Series)*, McGraw Hill, 1997.

Course Number Change - Computer Science Curriculum

Tanenbaum: *Distributed Operating Systems*, Prentice Hall, 1995.

Umar: *Object-Oriented Client/Server Internet Environments*, Prentice Hall, 1997.

Wang: *Java with Object-Oriented Programming and World Wide Web Applications*, ITP, 1999.

Yarger, et al: *MySQL and mSQL (Nutshell Series)*, O'Reilly & Associates, 1999.

Resources on the Web

Active Server Page Corner, <http://www.kamath.com/>

Client and Server Scripting in Web Pages, http://msdn.microsoft.com/library/techart/msdn_viscript.htm

Enterprise Java Beans and Java Server Pages <http://developer.java.sun.com/developer/onlineTraining/J2EE/Intro/index.html>

Java Tutorial Web Site <http://java.sun.com/docs/books/tutorial/>

Java Page by Dr. John Cross <http://www.iup.edu/~jacross/java.htmlx>

Robert Husted and J. J. Kuslich, *Server-Side JavaScript™: Developing Integrated Web Applications*,
<http://developer.netscape.com/tech/javascript/index.html?cp=dev01mtec>

TechMetrix report on "Internet Development Tools & Application Servers",
http://www.techmetrix.com/products/prod_report1.shtml (Tools evaluated: WebObjects 4.0, Sapphire/Web 5.1,
HAHTSite 3.1, Visual InterDev 6.0, NatWeb 2.0, Netscape App Server 2.1 Oracle App Server 4.0, Tango 3.5, PowerJ
Ent. 2.5, Apptivity 3.0, Silverstream 2.0)

TechMetrix report on "Java Application Servers", http://www.techmetrix.com/products/prod_report2.shtml (In addition to in-
depth product comparisons, this report explains the real potential and positioning of the latest Java technologies: EJB,
Java 2, Swing, ... ; Tools evaluated: Oracle, IBM, BEA, Inprise, Symantec, Gemstone)

XML Query Language (XQL), <http://www.w3.org/TandS/QL/QL98/pp/xql.html>

David T Smith

From: "Marcy Rearick" <MREARICK@iup.edu>
To: "David T Smith" <dtsmith@iup.edu>
Sent: Tuesday, January 10, 2006 11:33 AM
Subject: Re: COSC curriculum changes - confirmation of assignable number

Good Morning David,
Yes, it is okay to renumber COSC 344 back to 444 and
COSC 365 is an available course number.
Thanks for inquiring.
Marcy

----- Original Message -----

From: David T Smith
To: Rearick, Marcy M
Sent: Wednesday, January 04, 2006 12:45 PM
Subject: COSC curriculum changes - confirmation of assignable number

Ms. Rearick,

The Computer Science Curriculum Committee is in the process of update the curriculum. We are proposing a few changes to assigned numbers. Specifically:

Renumbering COSC 344 to COSC 444
Renumbering COSC 415 to COSC 365

Note the COSC 344 was originally COSC 444 and was changed a few years ago. We are desiring to renumber it back to its original number.

Can you confirm that i) renumbering COSC 344 back to COSC 444 is possible and will not cause problems, and ii) COSC 365 has not been previously assigned and can now be used in place of 415.

Thank-you in advance.

Best Regards,
David T. Smith