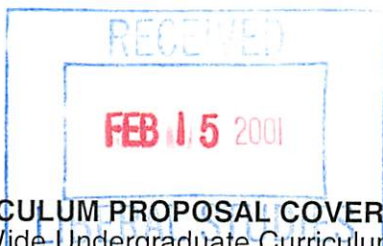


LSC Use Only
Number: _____
Submission Date: _____
Action-Date: _____



UWUCC USE Only
Number: _____
Submission Date: _____
Action-Date: _____

01-129
~~00-559~~
App. 4/17/01
Senate App 12/4/01

CURRICULUM PROPOSAL COVER SHEET
University-Wide Undergraduate Curriculum Committee

I. **CONTACT**

Contact Person T. Andrew Yang Phone x7-7995

Department Computer Science

II. **PROPOSAL TYPE (Check All Appropriate Lines)**

☒ **COURSE** Internet Arch and Prog
Suggested 20 character title

☒ New Course* CoSc 415 Internet Architecture and Programming
Course Number and Full Title

☐ Course Revision _____
Course Number and Full Title

☐ Liberal Studies Approval+
for new or existing course _____
Course Number and Full Title

☐ Course Deletion _____
Course Number and Full Title

☐ Number and/or Title Change _____
Old Number and/or Full Old Title

New Number and/or Full New Title

☐ Course or Catalog Description Change _____
Course Number and Full Title

☐ **PROGRAM:** _____ Major _____ Minor _____ Track

☐ New Program* _____
Program Name

☐ Program Revision* _____
Program Name

☐ Program Deletion* _____
Program Name

☐ Title Change _____
Old Program Name

New Program Name

III. **Approvals (signatures and date)**

Tia Watts
Department Curriculum Committee

[Signature]
College Curriculum Committee

+Director of Liberal Studies (where applicable)

[Signature]
Department Chair

[Signature]
College Dean

[Signature]
Provost (where applicable)



New Course
COSC 415 - Internet Architecture and Programming

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 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 client 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.

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 Javaspace*s, 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.

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_vscript.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>

Course Analysis
COSC 415 Internet Architecture and Programming

Section A: Details of the Course

- A1 This course is designed to enhance the Computer Science curriculum with an advanced Web-based development component, and is to be taken by junior or senior level Computer Science majors or students with equivalent training in Computer Science. Given the high demand for computer programmers who are capable of developing Internet-based software applications, this course will provide the Computer Science majors an opportunity to learn the state-of-the-art technology of Web development and architecture. Courses titled "Internet Programming" are being offered in many other colleges and universities, as well as in technology training institutes. Most of them, however, focus on a single computer language such as Java or Perl and do not cover the conceptual framework of Internet architecture and the rigorous process of Web-based software development. Nor do they cover advanced topics such as database connectivity or component-based software development. An advanced course covering both the fundamental Internet architecture and the software development methodologies is needed.
- A2 Two of the courses (COSC 341 and COSC 344) which are part of the pre-requisites of this course require change of their names, from COSC 441 to COSC 341 and COSC 444 to COSC 344 respectively. Proposals concerning these changes are submitted separately.
- A3 This course was offered as COSC 481 (Special Topics: Web-based information system) in Fall 1999. The offering was well received by Computer Science majors. The course is offered as COSC 481 (Special Topics: Internet Architecture and Programming) in Fall 2000.
- A4 The course is not intended to be offered as dual level.
- A5 The course may not be taken as variable credits.
- A6 Some colleges and universities have started to offer advanced courses such as this.

Allentown College in Center Valley, PA, for example, offers a course IT508 - INTERNET PROGRAMMING, which utilizes the Java language to introduce the concepts of object-oriented and internet programming, and covers advanced topics such as remote method invocation, database connectivity, and object request broker. (See <http://www4.allencol.edu/~dlm1/it508/class00/syllabus.html> for details.)

Another example is the City University at Hong Kong, which offers a Computer Science course CS4355 Internet Programming in Education that covers similar topics but with a focus on distance education software development (See <http://www.cityu.edu.hk/cityu/course/deptcurr/cscurr/cs4355.htm> for details.)

The Graduate Center of Marlboro College in Brattleboro, VT, offers a degree in Internet Engineering. One of the courses offered under the degree program is Internet Technologies, which covers an overview of Internetworking technologies, the software infrastructure of the Internet, the TCP/IP protocols, and pays particular attention to 'three- tier' architectures.

Although being offered to master and Ph.D. students in Computer Science and with a strong research focus, a course titled CSCI 6330: Advanced Internet Programming is offered by the Department of Computer Science, University of North Texas, Denton, Texas. The course covers advanced Internet programming techniques and Internet infrastructure and tools, with emphasis on e-commerce, Web-agents and Java based Web applications. (See <http://www.cs.unt.edu/~tarau/teaching/6330.html> for more details.)

- A7 The most recent curricular recommendation by the ACM (Association of Computing Machinery) was completed in 1991 (<http://www.acm.org/education/curricula.html>). The fact that Internet and World Wide Web did not become commercialized until 1994 explains why courses related to Internet and Web development was not incorporated into the 1991 curricular recommendation. The newest ACM curriculum recommendation was presented as a work in progress at ACM SIGCSE 2000. As noted by one of our colleagues in the department, the new recommendation addresses similar concerns to what are included in this syllabus of record. We anticipate courses on Internet programming and infrastructure will be incorporated into the next edition of the ACM recommendation.

Section B: Interdisciplinary Implications

- B1 The course is designed to be taught by one instructor.
- B2 This course requires substantial object-oriented programming and database application development background. It is offered to junior or senior standing Computer Science major students as a capstone course of our curriculum. There is no conflict with the offerings of other departments.
- B3 Students from the School of Continuing Education, if they possess the required pre-requisites, are welcome to take this course.

Section C: Implementation

- C1 Faculty resources are currently adequate. The COSC 315 (Large Files Processing) course is to be retired and partially replaced by this course.
- C2 Resources needed for this course are mostly available although they can be improved.
- a. Space: Classroom space is adequate.
 - b. Equipment: The course will use the desktop computers as the client workstations, and use both a Linux system and a Windows NT Server as the servers. Current equipment need to be updated in order to include a Windows NT Server 4.0 or above system running Microsoft Transaction Server and the Oracle Database Server. The Computer Science Department is currently seeking support in setting up the Windows NT Server. The equipment also needs to be maintained regularly in order to keep up with the demand of newer versions of software.
 - c. Laboratory Supplies and other Consumable Goods: This course requires the installation

of several types of computer programs, including the Web browsers, Web servers, database servers, development tools, and Web/database connectivity modules. Most of the software tools/packages run on the UNIX platform and are free for download. They do need to be maintained and updated continually.

- d. Library Materials: More reference books need to be ordered and added to the Stapleton Library. A list of the bibliography has been given to the Library.
- e. Travel Funds: Due to the complexity and the fast advancing nature of Internet architecture and technologies, travel funds are needed to sponsor the prospective instructor of this course in attending training classes and/or technical seminars to remain up-to-date in this field.

- C3 No resource for this course is funded by a grant.
- C4 The course is expected to be offered every semester.
- C5 It is anticipated that one or two sections of the course will be offered. If demand increases, more sections will be offered.
- C6 The enrollment in this course will be limited to the number of computers available in our teaching lab.
- C7 No professional society recommends enrollment limits or parameters for this course or for courses resembling this course. However, past experience has shown that twenty-five students per section can be accommodated based on the available seats in our teaching lab.