Course	Instructor	Count	Instructor Email	IIT/V and/or Internet
CS105-001/002/003	Hanrath	35	hanrath@iit.edu	N
CS105-004/005/006	Hanrath	54	hanrath@iit.edu	N
CS105 Night	Hanrath	27	hanrath@iit.edu	N
CS115	Koutsogiannakis	28	koutsogiannakis@iit.edu	N
CS116-001/002/007	Bauer	34	matthew.bauer@iit.edu	N
CS116-003/004	Winans	25	winans@iit.edu	N
CS116-005/006	Winans	16	winans@iit.edu	N
CS201	Sasaki	28	sasaki@iit.edu	N
CS330	Reingold	42	reingold@iit.edu	Y
MATH230	Ellis	32	ellisr@iit.edu	N
CS331	Beckman	45	beckman@iit.edu	Y
CS350	Hood	35	hood@iit.edu	N
CS351	Saelee	35	lee@iit.edu	Y
CS422	Goharian	28	nazli@ir.iit.edu	Y
CS425	Goharian	28	nazli@ir.iit.edu	Y
CS430	Kapoor	29	kapoor@iit.edu	Y
CS440	Beckman	17	beckman@iit.edu	Y
CS441	Koutsogiannakis	24	koutsogiannakis@iit.edu	Y
CS445	Bistriceanu	24	virgil@cs.iit.edu	Y
CS450 Day	Saelee	15	lee@iit.edu	N
CS450 Night	Saelee	24	lee@iit.edu	Y
CS455	Soneru	21	soneru@iit.edu	Y
CS458	Bistriceanu	9	virgil@cs.iit.edu	Y
CS480	Argamon	19	argamon@iit.edu	Y
CS485	Bauer, C	28	bauerc@iit.edu	N
CS487	Leung	28	leung@iit.edu	N
CS487 Rice	Aldawud	2	oaldawud@lucent.com	Y

TO: Faculty Teaching Undergraduate Computer Science Courses (Math230 included)
FROM: Computer Science Undergraduate Committee
DATE: April 14, 2008
SUBJECT: Computer Science Undergraduate Course Student Assessments

As part of ABET Accreditation and ongoing improvement to the Undergraduate Computer Science courses, the Computer Science Department requires Undergraduate Course Student Assessments to be done at the end of every semester.

- The assessments are to be completed during the last 2 weeks of the semester. The Computer Science Department expects 100% of its courses to be assessed by our students. If possible, announce one class in advance that the assessment will be conducted.
- Every different lecture section of a course should be assessed separately.
- The Course Manager of the course and you, the instructor of the course, will review these assessments AFTER course grades have been recorded.

Please read and follow the instructions below prior to initiating the assessment procedure for your class.

- Assessments are to be completed at the end of a single class period. No less than 15 minutes prior to the scheduled end of the class period, conclude class activities and begin the assessment process.
- Read the following statement to the class: "Please complete the assessment form to be passed out in order to provide feedback about this course section. We would like you to be thoughtful in your responses and to provide responsible, objective answers to the questions. These data and comments will be reviewed by the Computer Science Department and me, the instructor, after grades have been submitted."
- Request a volunteer to distribute the assessments, collect the assessments and seal them in this envelop, and drop the sealed envelop of assessments off in the Computer Science Department (room 236 Stuart Building).
- Immediately leave the classroom
- LATER, PLEASE VERIFY WITH MATTHEW BAUER at matthew.bauer@iit.edu THAT YOUR STUDENT ASSESSMENTS WERE RECEIVED BY THE CS DEPT.
- For courses with off-site students (IIT/V or internet sections) please attempt to send the "CS Course Student Assessment" to all off-site students via the IIT/V courier or via email (see www.cs.iit.edu/~abet for electronic copies of the assessment). Ask them to please return the completed assessment to the Computer Science Department (room 236 Stuart Building).

CS COURSE STUDENT ASSESSMENT

Course __CS105__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS105 - Introduction to Computer Programming I **Last Updated -** 01/25/02
Course Manager - Matthew Bauer, Senior Lecturer
2 credit hours; required for CS & CPE (or CS200); 100 min. lecture & 50 min. lab each week

Current Catalog Description - Introduces the use of a high-level programming language (C/C++) as a problem-solving tool—including basic data structures and algorithms, structured programming techniques, and software documentation. Designed for students who have had little or no prior experience with computer programming. (2-1-2)

Course Goals - Students should be able to:

1. Analyze and explain the behavior of simple programs involving the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, functions
2. Write a program that uses each of the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, functions
3. Break a problem into logical pieces that can be solved (programmed) independently.
4. Develop, and analyze, algorithms for solving simple problems.
5. Use a suitable programming language, and development environment, to implement, test, and debug algorithms for solving simple problems.
6. Write programs that use each of the following data structures (and describe how they are represented in memory): strings, arrays, and class libraries including strings and vectors

CS COURSE STUDENT ASSESSMENT

Course __CS115_ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS115– Object-Oriented Programming I **Last Updated - 1/29/04**

Course Manager - Matthew Bauer, Senior Lecturer

2 credit hours; required for CS & CPE (or CS201); 100 min. lecture & 50 min. lab each week

Catalog Description - Introduces the use of a high-level object-oriented programming language as a problem-solving tool – including basic data structures and algorithms, object-oriented programming techniques, and software documentation. Designed for students who have had little or no prior experience with computer programming. For students in CS and CS related degree programs. (2-1-2)

Course Goals - Students should be able to:

1. Analyze and explain the behavior of simple programs involving the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, methods
2. Write a program that uses each of the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, methods
3. Break a problem into logical pieces that can be solved (programmed) independently.
4. Develop, and analyze, algorithms for solving simple problems.
5. Use a suitable programming language, and development environment, to implement, test, and debug algorithms for solving simple problems.
6. Write programs that use each of the following data structures (and describe how they are represented in memory): strings, arrays
7. Explain and apply object-oriented design and testing involving the following concepts: data abstraction, encapsulation, information hiding
8. Use a development environment to design, code, test, and debug simple programs, including multi-file source projects, in an object-oriented programming language
9. Implement basic error handling.
10. Apply appropriate problem-solving strategies.
11. Use APIs (Application Programmer Interfaces) and design/program APIs

CS COURSE STUDENT ASSESSMENT

Course __CS116__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

8. Did you understand what was expected of you in the course? Yes / No

9. Do you think that you achieved the course goals (see back) for the course? Yes / No

10. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
    _____
    _____
    _____

11. Were you adequately prepared to take this course by your mathematics background? Yes / No

12. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
    Yes / No / Not applicable

13. What did you like best about this course?
    _____
    _____
    _____
    _____
    _____

14. What, if anything, would you change about this course?
    _____
    _____
    _____
    _____
    _____

Please continue on the back if necessary.

CS116 – Object-Oriented Programming II **Last Updated - 1/29/04**

Course Manager - Matthew Bauer, Senior Lecturer

2 credit hours; required for CS & CPE (or CS201); 75 min. lecture & 75 min. lab each week

Catalog Description - Continuation of CS 115. Introduces more advanced elements of object-oriented programming – including dynamic data structures, recursion, searching and sorting, and advanced object-oriented programming techniques. For students in CS and CS related degree programs. Prerequisite: CS 115 (2-1-2)

Course Goals - Students should be able to:

1. Analyze and explain the behavior of simple programs involving the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, methods
2. Write a program that uses each of the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, methods
3. Break a problem into logical pieces that can be solved (programmed) independently.
4. Develop, and analyze, algorithms for solving simple problems.
5. Use a suitable programming language, and development environment, to implement, test, and debug algorithms for solving simple problems.
6. Write programs that use each of the following data structures (and describe how they are represented in memory): strings, arrays
7. Explain the basics of the concept of recursion.
8. Write, test, and debug simple recursive functions and procedures.
9. Explain and apply object-oriented design and testing involving the following concepts: data abstraction, encapsulation, information hiding
10. Use a development environment to design, code, test, and debug simple programs, including multi-file source projects, in an object-oriented programming language
11. Implement basic error handling.
12. Solve problems by creating and using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)
13. Determine the time complexity of simple algorithms.
14. Apply appropriate problem-solving strategies.
15. Use APIs (Application Programmer Interfaces) and design/program APIs

CS COURSE STUDENT ASSESSMENT

Course __CS201__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

15. Did you understand what was expected of you in the course? Yes / No

16. Do you think that you achieved the course goals (see back) for the course? Yes / No

17. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
_____
_____
_____

18. Were you adequately prepared to take this course by your mathematics background? Yes / No

19. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
Yes / No / Not applicable

20. What did you like best about this course?
_____
_____
_____
_____
_____

21. What, if anything, would you change about this course?
_____
_____
_____
_____
_____

Please continue on the back if necessary.

CS201 - Accelerated Introduction to Computer Science **Last Updated -** 9/1/03
Course Manager - Matthew Bauer, Senior Lecturer
4 credit hours; required for CS & CPE (or CS115/CS116); 150 min. lecture & 100 min. lab each week

Current Catalog Description - Problem-solving and design using an object-oriented programming language. Introduces a variety of problem solving techniques, algorithms, and data structures in object-oriented programming. Prerequisites: CS105 or CS 115 or experience using any programming language. (3-2-4)

Course Goals - Students should be able to:

1. Analyze and explain the behavior of simple programs involving the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, methods
2. Write a program that uses each of the following fundamental programming constructs: assignment, I/O (including file I/O), selection, iteration, methods
3. Break a problem into logical pieces that can be solved (programmed) independently.
4. Develop, and analyze, algorithms for solving simple problems.
5. Use a suitable programming language, and development environment, to implement, test, and debug algorithms for solving simple problems.
6. Write programs that use each of the following data structures (and describe how they are represented in memory): strings, arrays
7. Explain the basics of the concept of recursion.
8. Write, test, and debug simple recursive functions and procedures.
9. Explain and apply object-oriented design and testing involving the following concepts: data abstraction, encapsulation, information hiding
10. Use a development environment to design, code, test, and debug simple programs, including multi-file source projects, in an object-oriented programming language
11. Implement basic error handling.
12. Solve problems by creating and using sequential search, binary search, and quadratic sorting algorithms (selection, insertion)
13. Determine the time complexity of simple algorithms.
14. Apply appropriate problem-solving strategies.
15. Use APIs (Application Programmer Interfaces) and design/program APIs
16. Time permitting: basic graphical user interface (GUI), event-driven programming

CS COURSE STUDENT ASSESSMENT

Course __CS330 or MATH230__ Semester ___SPRING 2008___ Grad or Undergrad
(circle one)

INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS330 – Discrete Structures **Last Updated -** 02/19/02

Course Manager - Dr. Sanjiv Kapoor, Professor

3 credit hours; required for CS & CPE (or MATH230); 150 min. lecture each week

Current Catalog Description - Introduction to the use of formal mathematical structures to represent problems and computational processes. Topics covered include Boolean algebra, first-order logic, recursive structures, graphs, and abstract language models. Prerequisite: CS 106 or CS 200. (3-0-3)

Course Goals - Students should be able to:

1. Illustrate by examples the basic terminology of functions, relations, and sets and demonstrate knowledge of their associated operations.
2. Demonstrate in practical applications the use of basic counting principles of permutations, combinations, inclusion/exclusion principle and the pigeonhole methodology.
3. Calculate probabilities of events and expectations of random variables for problems arising from games of chance.
4. Establish and solve recurrence relations that arise in counting problems including the problem of determining the time complexity of recursively defined algorithms.
5. Model logic statements arising in algorithm correctness and real-life situations and manipulate them using the formal methods of propositional and predicate logic.
6. Outline basic proofs for theorems using the techniques of - direct proofs, proof by counterexample, proof by contraposition, proof by contradiction, mathematical induction.
7. Relate the ideas of mathematical induction to recursion and recursively defined structures.
8. Illustrate by example basic terminology of graph theory and model problems in computer science using graphs and trees.
9. Deduce properties that establish particular graphs as Trees, Planar, Eulerian, and Hamiltonion.
10. Illustrate the application of trees and graphs to data structures.
11. Explain the basic concepts modeling computation including formal machines, languages, finite automata, Turing machines

CS COURSE STUDENT ASSESSMENT

Course __CS331__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS331 – Data Structures and Algorithms **Last Updated -** 02/06/02
Course Manager - Dr. Gruia Calinescu, Assistant Professor
3 credit hours; required for CS & CPE; 100 min. lecture & 100 min. lab each week

Current Catalog Description - Implementation and application of the essential data structures used in computer science. Analysis of basic sorting and searching algorithms and their relationship to these data structures. Particular emphasis is given to the use of object-oriented design and data abstraction in the creation and application of data structures. Prerequisite: CS 106 or CS 200. (2-2-3)

Course Goals - Students should be able to:

1. Explain, implement, and apply the following data-structures: lists (unordered and ordered), stacks, queues, expression trees, binary search trees, heaps, and hash tables.
2. Analyze the time and space complexity of algorithms using asymptotic upper bounds (big-O notation).
3. Explain and use references and linked structures.
4. Outline basic object-oriented design concepts: composition, inheritance, polymorphism.
5. Write and test recursive procedures, and explain the run-time stack concept.
6. Analyze searching and sorting algorithms, and explain their relationship to data-structures.
7. Choose and implement appropriate data-structures to solve an application problem.
8. Explain how to use unit tests and version control in your software development.

CS COURSE STUDENT ASSESSMENT

Course __CS350__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS350 -Computer Organization and Assembly Language Programming Last Updated - 01/01/05
Course Manager – Dr. Cindy Hood, Associate Professor
3 credit hours; required for CS & CPE (or ECE242 allowed for CPE); 100 min. lecture & 100 min. lab each week

Current Catalog Description - Introduction to the internal architecture of computer systems - including micro, mini-, and mainframe computer architectures. Focuses on the relationship between a computer's hardware, its native instruction set, and the implementation of high-level languages on that machine. Uses a set of assembly language programming exercises to explore and analyze microcomputer architecture. Prerequisite: CS 106 or CS 200. (2-2-3) (C)

Course Goals - Students should be able to:

1. Explain the layers of abstraction an overview of computer systems.
2. Develop and debug low-level programs in C including pointers and dynamic memory allocation.
3. Explain and solve problems about data representation in computers including:
    - Number systems and Boolean algebra
    - Unsigned, Two's complement, Floating point
    - Limitations of electronic circuits
    - Arithmetic
4. Write and debug assembly language programs (IA32) and explain the following implementation details:
    - ISA design
    - Compilers and assemblers
    - Translating HLL control constructs
    - Complex data structures
5. Explain the basics of processor architecture including:
    - Digital logic and HDLs
    - Basic datapath/control model
    - Pipelining overview
6. Explain the concepts of performance optimization including:
    - Capabilities of optimizing compilers
    - Machine independent program transformations
    - Machine dependent optimizations
7. Explain Memory Hierarchy including:
    - Memory hierarchy overview
    - Locality of reference
    - Caching methodologies
    - Optimizing program performance with improved locality
8. Explain the linking process including:
    - Understanding role of linking in compilation
    - Static and dynamic linking

CS COURSE STUDENT ASSESSMENT

Course __CS351__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS351 – Systems Programming **Last Updated -** 0101/05
Course Manager – Matthew Bauer, Senior Lecturer
3 credit hours; required for CS & CPE; 100 min. lecture & 100 min. lab each week

Current Catalog Description - Examines the components of sophisticated multi-layer software systems-including device drivers, systems software, applications interfaces, and user interfaces. Explores the design and development of interrupt-driven and event-driven software. Prerequisites: CS 331, CS 350. (2-2-3)

Course Goals - Students should be able to:

1. Define the concept and role of a process in a modern operating system
2. Describe the key abstractions an operating system provides to running processes
3. Describe the function, usage, and operation of system calls related to process management, memory management and I/O
4. Explain exceptional control flow, including:
    - Hardware interrupts
    - Software exceptions / Traps
    - Signals and signal handling
5. Describe the essential operation of a modern MMU from a programmer's standpoint, including:
    - Caching and the TLB
    - Segmentation and paging for virtual memory
6. Explain the operation of various memory allocation methods, including:
    - Implicit allocation (garbage collection)
    - Explicit allocation (malloc/free, reference counting, etc.)
7. Describe, utilize, and implement a dynamic memory allocation API.
8. Describe and utilize the system-level I/O API of a modern operating system, including:
    - File descriptors
    - File I/O
    - Buffered I/O
    - Interprocess communication
9. Describe and utilize a low-level socket based networking API. This should include:
    - Client / Server model
    - Internetworking
    - Berkeley sockets
10. Describe, design and utilize concurrent programming APIs, including:
    - POSIX Threads
    - Re-Entrant code
    - Synchronization primitives

CS COURSE STUDENT ASSESSMENT

Course __CS422__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).

_____
_____
_____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
Yes / No / Not applicable

6. What did you like best about this course?

_____
_____
_____
_____
_____

7. What, if anything, would you change about this course?

_____
_____
_____
_____
_____

Please continue on the back if necessary.

CS422 – Introduction to Data Mining **Last Updated -** 04/25/03
Course Manager – Dr. Nazli Goharian, Clinical Assistant Professor
3 credit hours; elective for CS & CPE; 150 min. lecture each week

Current Catalog Description - This course will provide an introductory look at concepts and techniques in the field of data mining. After covering the introduction and terminologies to Data Mining, the techniques used to explore the large quantities of data for the discovery of meaningful rules and knowledge such as market basket analysis, nearest neighbor, decision trees, neural networks, and clustering are covered. The students learn the material by implementing different techniques throughout the semester (3-0-3). Prerequisites: CS 331 or CS 401 or CS 403 and strong programming knowledge.

Course Goals - Students should be able to:

1. Explain the Data Mining motivation and applications.
2. Explain the Data Mining Architecture.
3. Explain Data Preprocessing motivation and techniques.
4. Explain various Data Mining algorithms such as Naïve Bayes, Neural Networks, Decision Tree, Association-Rules, and Clustering.
5. Explain the scalability issues for each of the algorithms discussed in the class and how they can be modified for scalability.
6. Design and implement data mining systems using various data pre-processing techniques and mining algorithms.
7. Apply the research ideas into their experiments in building data mining systems.

CS COURSE STUDENT ASSESSMENT

Course __CS425__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).

_____
_____
_____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
Yes / No / Not applicable

6. What did you like best about this course?

_____
_____
_____
_____
_____

7. What, if anything, would you change about this course?

_____
_____
_____
_____
_____

Please continue on the back if necessary.

CS425 – Database Organization **Last Updated -** 02/22/02
Course Manager – Dr. Nazli Goharian, Clinical Assistant Professor
3 credit hours; elective for CS & CPE; 150 min. lecture each week

Current Catalog Description - Overview of database architectures, including the Relational, Hierarchical, Network, and Object Models. Database interfaces, including the SQL query language. Database design using the Entity-Relationship Model. Issues such as security, integrity, and query optimization. Prerequisite: CS 331 or CS 401 or CS 403. (3-0-3) (T)

Course Goals - Students should be able to:

1. Design and model a design scenario using relational data modeling, which includes:
    a. Analyze the design anomalies
    b. Construct Entity Relationship Diagram.
    c. Analyze and Construct Functional Dependencies for the business rules.
    d. Analyze Functional Dependencies to identify Primary keys.
    e. Analyze and Perform Normalization and Normal Forms.
    f. Define referential integrities.
    g. Create relational database design schemas in 3-NF for a design scenario of the size of ca. 8-10 tables.
2. Solve abstract relational language, such as relational algebra problems.
3. Solve database transactions by using Structured Query Language (SQL), used by commercial RDBMSs.
4. Explain File Organizations, Indexing, and Query Processing.
5. Explain Query Optimizations such as Rule-Based and Cost-Based.
6. Explain Concurrency Control.
7. Explain Recovery.
8. Implement a relational database application, using a commercial RDBMS.

CS COURSE STUDENT ASSESSMENT

Course __CS430__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS430 - Introduction to Algorithms **Last Updated -** 03/01/02
Course Manager – Dr. Sanjiv Kapoor, Professor
3 credit hours; required for CS, elective for CPE; 150 min. lecture each week

Current Catalog Description - Introduction to the design, behavior, and analysis of computer algorithms. Searching, sorting, and combinatorial algorithms are emphasized. Worst case and average bounds on time and space usage. Prerequisites: (CS 330 and CS 331) or CS401 or CS403.

Course Goals - Students should be able to:

1. Use big O, omega, and theta notation to give asymptotic upper, lower, and tight bounds on time and space complexity of algorithms.
2. Determine the time complexity of simple algorithms, deduce the recurrence relations that describe the time complexity of recursively defined algorithms, and solve simple recurrence relations. Also apply amortized analysis.
3. Design algorithms using the brute-force, greedy, dynamic programming, divide-and-conquer, branch and bound strategies.
4. Design algorithms using at least one other algorithmic strategy from the list of topics for this unit.
5. Use and implement the fundamental abstract data types -- specifically including balanced binary search trees, and graphs -- necessary to solve algorithmic problems efficiently.
6. Solve problems using techniques learned in the design of sequential search, binary search, O(N log N) sorting algorithms, order statistics and fundamental graph algorithms, including depth-first and breadth-first search, single-source and all-pairs shortest paths, and at least one minimum spanning tree algorithm.
7. Demonstrate the following abilities: to evaluate algorithms, to select from a range of possible options, to provide justification for that selection, and to implement the algorithm in simple programming contexts.
8. Communicate theoretical and experimental analyses of a set of algorithms (i.e. sorting) in a lab report format.

CS COURSE STUDENT ASSESSMENT

Course __CS440__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS440 -Programming Languages and Translators (3 credit hours) **Last Updated -** 03/11/02
Course Manager – Dr. Xiang-Yang Li, Assistant Professor
3 credit hours; required for CS, elective for CPE; 150 min. lecture each week

Current Catalog Description - Study of commonly used computer programming languages with an emphasis on precision of definition and facility in use. Scanning, parsing, and introduction to compiler design. Use of compiler generating tools. Prerequisite: CS 330, CS 351. (3-0-3)

Course Goals - Students should be able to:

1. Explain major classes of programming languages: techniques, features, and styles.
    - Know how to use boxed and unboxed variables
    - Be able to use higher order functions.
2. How to specify formally the meaning of a language --- to people and to the computer.
    - Use Transition, Typing, and Denotational Semantics to define a language construct.
    - Be able to specify the language of regular expressions.
    - Determine if a grammar is LL, and write a parser for it using recursive descent.
    - Determine if a grammar is LR, and write a parser for it using a parser generator.
    - Describe the algorithm for both LL and LR parser generation.
3. Explain Three Powerful Ideas:
    - Recursion
        - Know how to use both tail recursion and standard recursion.
        - Know how to use higher order functions to eliminate recursion.
    - Abstraction
        - Know how to create user-defined types.
        - Know how to use functions to model integers.
        - Know how to use trees to model language constructs.
    - Transformation
        - Know how to interpret a language.
        - Know how to use unification.
4. How to choose a language.
5. How to implement a language.
6. Emphasis: learn theory and apply it.

CS COURSE STUDENT ASSESSMENT

Course __CS441__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS441 - Current Topics in Programming Languages **Last Updated -** 04/15/02
Course Manager – Dr. Tzilla Elrad, Research Professor
3 credit hours; elective for CS & CPE; 150 min. lecture each week

Current Catalog Description – New topics in programming language design such as concepts of concurrent and distributed programming, communicating sequential processes, and functional programming. System development tools and language features for programming. An introduction to programming language semantics. Prerequisite: CS 331 or CS 401 or CS 403. (3-0-3)

Course Goals
The course is basically language independent. Any language that can support the course goals may be selected. An example of a choice language might be Java.

1. Outline the evolution of the architectural neutral, secure, OO programming languages in order to illustrate how this evolution has led to the occurrence of the JAVA programming model. The course builds on the students ' knowledge of Object Oriented Programming concepts, which is a prerequisite for the course.
2. Design, implement, test, and debug Applets, Servlets, and Applications in the JAVA language
3. Design and implement Graphical User Interfaces using Java's Swing components and Java's AWT classes, interfaces and methods.
4. Learn the programming language mechanisms that support distribution transparency and development of distributed applications; JAVA sockets and RMI.
5. Recognize the underlying concurrency language model; Multithreading and JAVA monitor-based concurrency model.
6. Demonstrate the supportive language constructs and mechanisms for the design and development of 3-tier architectures; JDBC and JAVA Servlets for server-side programming.

CS COURSE STUDENT ASSESSMENT

Course __CS445__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS445 - Object Oriented Design and Programming **Last Updated -** 04/16/02
Course Manager – Dr. Bogdan Korel, Associate Professor
3 credit hours; elective for CS & CPE; 150 min. lecture

Current Catalog Description - Introduction to methodologies for object-oriented design and programming. Examines the object model and how it is realized in various object-oriented languages. Focuses on methods for developing and implementing object-oriented systems. Prerequisite: CS 331 or CS 401 or CS 403 (3-0-3)

Course Goals - Students should be able to:

1. Explain and justify the principles of Object Oriented concepts (review abstraction & abstract data types, encapsulation, inheritance, polymorphism, aggregation)
2. Analyze and identify the strengths (and weaknesses) of in-depth areas of the Object Oriented paradigm.
3. Analyze, explain, & compare the qualities of Object Oriented languages and how well they support the object model.
4. Explain and analyze the key points of Object Oriented analysis.
5. Explain and analyze the key points of Object Oriented design.
6. Design, implement, test and debug multi-phased Object Oriented application.
7. Utilize contemporary Object Oriented methodology and notation to express the products of Object Oriented Analysis & Design (class design, class relationships, object interaction, object states, etc)
8. Perform Object Oriented Analysis & Design on a real-world problem.
9. Explain and Utilize Complex Design Patterns.
10. Create an implementation of the resultant Object Oriented design.
11. Examine new & contemporary concepts in Object Orientation.
12. Communicate the deliverables of a software development project.

CS COURSE STUDENT ASSESSMENT

Course __CS450__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS450 - Introduction to Operating Systems **Last Updated -** 04/02/02
Course Manager – Dr. Xian-He Sun, Associate Professor
3 credit hours; required for CS & CPE; 150 min. lecture each week

Current Catalog Description - Introduction to operating system concepts—including system organization for uniprocessors and multiprocessors, scheduling algorithms, process management, deadlocks, paging and segmentation, files and protection, and process coordination and communication. Prerequisites: (CS 331 and CS 350) or (CS 331 and ECE 242) or (CS 401 and CS 402) or CS 403. (3-0-3)

Course Goals - Students should be able to:

1. Explain the range of requirements that a modern operating system has to address.
2. Define the functionality that a modern operating system must deliver to meet a particular need.
3. Articulate design tradeoffs inherent in operating system design.
4. Explain the concept of a logical layer.
5. From the perspective of building operating systems, explain the benefits of building these layers in a hierarchical fashion.
6. Describe how the resources of the computer system are managed by software.
7. Relate system state to user protection.
8. Justify the presence of concurrency within the framework of an operating system.
9. Demonstrate the potential run-time problems arising from the concurrent operation of many (possibly a dynamic number of) tasks.
10. Summarize the range of mechanisms (at an operating system level) that can be employed to realize concurrent systems and be able to describe the benefits of each.
11. Explain the different states that a task may pass through and the data structures needed to support the management of many tasks.
12. Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems.
13. Describe relationships between scheduling algorithms and application domains.
14. Investigate the wider applicability of scheduling in such contexts as disk I/O, networking scheduling, and project scheduling.
15. Introduce memory hierarchy and cost-performance tradeoffs.
16. Explain what virtual memory is and how it is realized in hardware and software.
17. Examine the wider applicability and relevance of the concepts of virtual entity and of caching.
18. Evaluate the trade-offs in terms of memory size (main memory, cache memory, auxiliary memory) and processor speed.
19. Defend the different ways of allocating memory to tasks on the basis of the relative merits of each.
20. Summarize the features of an operating system used to provide protection and security, and describe the limitations of each of these.
21. Summarize the full range of considerations that support file systems.

CS COURSE STUDENT ASSESSMENT

Course __CS455__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS455 - Data Communications **Last Updated -** 03/20/02

Course Manager – Dr. Peng-Jun Wan, Assistant Professor

3 credit hours; elective for CS & CPE; 150 min. lecture each week

Current Catalog Description - Introduction to data communication concepts and facilities with an emphasis on protocols and interface specifications. Focuses on the lower four layers of the ISO-OSI reference model. Prerequisite: CS 450. (3-0-3)

Course Goals - Students should be able to:

1. Explain the operation of multi-layered protocols, particularly the OSI and Internet models / architectures and how standards evolve.
2. Describe the difference between different network topologies, including packet and circuit switched, LANs and WANs, and identify and describe networks that apply to each network type.
3. Explain the basic concepts of the Physical Layer: including physical media, encoding / modulation, multiplexing, error control, and their implementation in various commercial networks.
4. Describe the basic operation of the Data Link Layer, including connection oriented versus connectionless protocols, retransmission algorithms, windows and flow control, and their implementations in various networks.
5. Describe the basic operation of the network layer, including addressing and routing.
6. Describe the basic operation of TCP/UDP, including connection establishment and release, buffered transfer, adaptive retransmission, and congestion and flow control.
7. Describe LAN architectures and their implementations.
8. Explain Application layer concepts, including commercial Internet protocols and client-server technologies. Explain special issues, including security, performance, and quality of service from a technical and ethical viewpoint.
9. Tie in all above concepts to describe the global data / telecommunications network.

CS COURSE STUDENT ASSESSMENT

Course __CS458__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).

_____
_____
_____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
Yes / No / Not applicable

6. What did you like best about this course?

_____
_____
_____
_____
_____

7. What, if anything, would you change about this course?

_____
_____
_____
_____
_____

Please continue on the back if necessary.

CS458 – Information Security **Last Updated -** 02/03/04
Course Manager – David Grossman, Associate Professor
3 credit hours; elective for CS & CPE; 150 min. lecture each week

Current Catalog Description: Prerequisites: CS 425 and CS 450. Three hour lecture. An introduction to the fundamentals of computer and information security. This course focuses on algorithms and techniques used to defend against malicious software. Topics include an introduction to encryption systems, operating system security, database security, network security, system threats, and risk avoidance procedures.

Course Goals - Students should be able to:

1. Provide an introduction to the security engineering discipline
2. Expose students to contemporary risks and attack procedures.
3. To provide students with an appreciation of the historical perspective in information assurance research.
4. Describe security engineering processes – particularly those being used in industry.
5. Students will be familiar with fundamental encryption algorithms
6. Students will be able to design an architecture to defend a specific system from attack.
7. The student will be able to apply standard, accepted security engineering techniques to protect a system with respect to a specific organizational security policy.
8. The student will demonstrate an ability to document their work to an acceptable standard.

CS COURSE STUDENT ASSESSMENT

Course __CS480__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS480 - Artificial Intelligence: Planning and Control **Last Updated -** 02/07/07
Course Manager - Dr. Shlomo Argamon, Associate Professor
3 credit hours; elective for CS & CPE; 150 min. lecture each week

Current Catalog Description - Introduction to computational methods of intelligent control of autonomous agents, and the use of programming paradigms that support development of flexible and reactive systems. These include heuristic search, knowledge representation, constraint satisfaction, probabilistic reasoning, decision-theoretic control, and sensor interpretation. Particular focus will be places on real-world application of the material. (3-0-3). Prerequisite: CS 331 or CS 401 or CS 403. Corequisite: MATH 474 or equivalent. (3-0-3) (T)

Course Goals - Students should be able to:

- Describe the Turing test.
- Explain the concepts of optimal reasoning, human-like reasoning, optimal behavior, human-like behavior.
- Develop "PAGE" descriptions of an agents and determine which agent type is applicable to a problem.
- Solve problems in a functional programming language (LISP)
- Formulate an efficient problem space for a problem expressed in English by expressing that problem space in terms of states, operators, an initial state, and a description of a goal state.
- Describe the problem of combinatorial explosion and its consequences.
- Select an appropriate brute-force search algorithm for a problem, implement it, and characterize its time and space complexities.
- Select an appropriate heuristic search algorithm for a problem and implement it by designing the necessary heuristic evaluation function.
- Describe under what conditions heuristic algorithms guarantee optimal solution.
- Implement minimax search with alpha-beta pruning for some two-player game.
- Formulate a problem specified in English as a constraint-satisfaction problem and implement it using a chronological backtracking algorithm.
- Explain the operation of the resolution technique for theorem proving.
- Apply Bayes theorem to determine conditional probabilities.
- Explain the distinction between monotonic and non-monotonic inference.
- Explain the differences among the three main styles of learning: supervised, reinforcement, and unsupervised.
- Implement simple algorithms for supervised learning, reinforcement learning, and unsupervised learning.
- Determine which of the three learning styles is appropriate to a particular problem domain.
- Compare and contrast each of the following techniques, providing examples of when each strategy is superior: decision trees, neural networks, and belief networks. Explain the nearest neighbor algorithm and its place within learning theory.

CS COURSE STUDENT ASSESSMENT

Course __CS485__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS485 – Computers and Society **Last Updated -** 03/08/02

Course Manager - Charles Bauer, Professor Emeritus

3 credit hours; required for CS, not allowed for CPE; 150 min. lecture each week

Current Catalog Description - Discussion of the impact of computer technology on present and future society. Historical development of the computer. Social issues raised by cybernetics. Prerequisites: COM421 or COM428. (3-0-3) (C)

Course Goals - Students should be able to:

1. Demonstrate an understanding of the social and professional context in which computing is done.
2. Demonstrate an understanding of the basic cultural, social, legal, and ethical issues inherent in the discipline of computing.
3. Identify milestones in the development and application of information technology.
4. Ask serious questions about the social impact of computing and to evaluate proposed answers to those questions.
5. Demonstrate an awareness of the basic legal rights of software and hardware vendors and users, and the ethical values that are the basis for those rights.
6. Research the social and ethical issues of a computer related topic from the list in the syllabus.
7. Communicate, both orally and in written form, social and ethical issues of a computer related topic from the list in the syllabus.

CS COURSE STUDENT ASSESSMENT

Course __CS487__ Semester ___SPRING 2008___ Grad or Undergrad (circle one)
INSTRUCTOR: _____ SECTION: _____

The CS department is committed to continuous improvement of its programs and would like to have more information than is provided by the standard IIT course evaluation form. Please take a few minutes to complete this survey. Thank you.

1. Did you understand what was expected of you in the course? Yes / No

2. Do you think that you achieved the course goals (see back) for the course? Yes / No

3. If you answered "No" to question 2, which goal or goals did you not achieve (please list by number from back).
   _____
   _____
   _____

4. Were you adequately prepared to take this course by your mathematics background? Yes / No

5. Were you adequately prepared to take this course by prerequisite computer science courses (see back)?
   Yes / No / Not applicable

6. What did you like best about this course?
   _____
   _____
   _____
   _____
   _____

7. What, if anything, would you change about this course?
   _____
   _____
   _____
   _____
   _____

Please continue on the back if necessary.

CS487 - Software Engineering **Last Updated -** 04/11/02

Course Manager – Dr. Ilene Burnstein, Associate Professor

3 credit hours; required for CS & CPE; 150 min. lecture each week

Current Catalog Description - Study of the principles and practices of software engineering. Topics include software quality concepts, process models, software requirements analysis, design methodologies, software testing, and software maintenance. Hands-on experience building a software system using the waterfall life cycle model. Students working in teams develop all life cycle deliverables: requirements document, specification and design documents, system code, test plan, and user manuals. Prerequisite: CS 331 or CS 401 or CS 403. (3-0-3) (C)

Course Goals - Students should be able to:

1. Understand and explain software development as a series of engineering activities, and processes.
2. Demonstrate software development team-working skills.
3. Analyze client/user needs.
4. Select an appropriate life cycle and process model for development of a software product.
5. Explain the importance of software quality evaluation activities.
6. Develop a series of software life-cycle deliverables.
7. Develop representations/models and descriptions of an evolving software product for inclusion in a requirements specification document.
8. Build a multi-level design model and evaluate software design alternatives
9. Design, execute, and log multi-level software tests.
10. Describe the role that tools can play in the software life cycle.
11. Communicate, verbally and in writing, the deliverables of a software development project.