

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS105_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS115_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS116_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____
 INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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, inheritance, polymorphism				
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS201_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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, inheritance, polymorphism				
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				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS330/MATH230_ Semester: _SPRING 2008_ # of students (live): ____ (remote): ____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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 Hamiltonian.				
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				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS331_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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 time and space complexity of algorithms using asymptotic upper bounds (the big-O notation)				
3. Explain and use references and linked structures.				
4. Outline basic object-oriented design concepts: composition, inheritance, and polymorphism				
5. Write and test recursive procedures with linked structures, and explain the run-time stack concept				
6. Analyze sorting and searching 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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS350_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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: <ul style="list-style-type: none"> ○ 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: <ul style="list-style-type: none"> ○ ISA design ○ Compilers and assemblers ○ Translating HLL control constructs ○ Complex data structures 				
5. Explain the basics of processor architecture including: <ul style="list-style-type: none"> ○ Digital logic and HDLs ○ Basic datapath/control model ○ Pipelining overview 				
6. Explain the concepts of performance optimization including: <ul style="list-style-type: none"> ○ Capabilities of optimizing compilers ○ Machine independent program transformations ○ Machine dependent optimizations 				
7. Explain Memory Hierarchy including: <ul style="list-style-type: none"> ○ Memory hierarchy overview ○ Locality of reference ○ Caching methodologies ○ Optimizing program performance with improved locality 				
8. Explain the linking process including: <ul style="list-style-type: none"> ○ Understanding role of linking in compilation ○ Static and dynamic linking 				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS351_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS422_ Semester: _SPRING 2003_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS425_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS430_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS440_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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 Recursion: Know how to use both tail recursion and standard recursion. Know how to use higher order functions to eliminate recursion.				
4. Explain 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.				
5. Explain Transformation: Know how to interpret a language. Know how to use unification.				
6. How to choose a language.				
7. How to implement a language.				
8. Emphasis: learn theory and apply it.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS441_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS445_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS450_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS455_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS458 _____ SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS480_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
1. Describe the Turing test.				
2. Explain the concepts of optimal reasoning, human-like reasoning, optimal behavior, human-like behavior.				
3. Develop "PAGE" descriptions of an agents and determine which agent type is applicable to a problem.				
4. Solve problems in a functional programming language (LISP)				
5. 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.				
6. Describe the problem of combinatorial explosion and its consequences.				
7. Select an appropriate brute-force search algorithm for a problem, implement it, and characterize its time and space complexities.				
8. Select an appropriate heuristic search algorithm for a problem and implement it by designing the necessary heuristic evaluation function.				
9. Describe under what conditions heuristic algorithms guarantee optimal solution.				
10. Implement minimax search with alpha-beta pruning for some two-player game.				
11. Formulate a problem specified in English as a constraint-satisfaction problem and implement it using a chronological backtracking algorithm.				
12. Explain the operation of the resolution technique for theorem proving.				
13. Apply Bayes theorem to determine conditional probabilities.				
14. Explain the distinction between monotonic and non-monotonic inference.				
15. Explain the differences among the three main styles of learning: supervised, reinforcement, and unsupervised.				
16. Implement simple algorithms for supervised learning, reinforcement learning, and unsupervised learning.				
17. Determine which of the three learning styles is appropriate to a particular problem domain.				
18. 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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS485_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)

CS FACULTY/COURSE SELF-ASSESSMENT

Course: CS487_ Semester: _SPRING 2008_ # of students (live): _____ (remote): _____

INSTRUCTOR: _____ SECTIONS: _____

1. How many students earned each grade?

(live) A _____ B _____ C _____ D _____ E _____

(remote) A _____ B _____ C _____ D _____ E _____

2. Indicate how many students met each course goal by placing an X in each row for each Course Goal.

COURSE GOALS	ALL	MOST	SOME	NONE
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.				

3. In your opinion, were the students adequately prepared in mathematics? (Yes/No). If "No", what deficiencies did you identify? _____

4. In your opinion, were the students adequately prepared in computer science basics? (Yes/No). If "No", what deficiencies did you identify? _____

5. In your opinion, were the students adequately prepared by other pre-requisite courses, if any? (Yes/No). If "No", what deficiencies did you identify? _____

6. If you were to teach the course again, what changes would you make? (Use additional pages if needed)
