End of Course Memo
CS 202 – Discrete Math
Aaron Bloomfield (Spring 2005)

Course Objectives:

1. Logic: Introduce a formal system (propositional and predicate logic) which mathematical reasoning is based on.
2. Proofs: Develop an understanding of how to read and construct valid mathematical arguments (proofs) and understand mathematical statements (theorems), including inductive proofs. Also, introduce and work with various problem solving strategies and techniques.
3. Counting: Introduce the basics of integer theory, combinatorics, and counting principles, including a brief introduction to discrete probability.
4. Structures: Introduce and work with important discrete data structures such as sets, relations, sequences, and discrete functions.
5. Applications: Gain an understanding of some application areas of the material covered in the course.

Assessment of Learning by Course-Objective:

For the assessment of these objectives, I analyze the final exam, which was split into 5 parts to correspond to the various course objectives. The final was difficult, but fair. The overall average on the final exam was 80.7%.

Objective 1: Logic: Introduce a formal system (propositional and predicate logic) which mathematical reasoning is based on.
The final exam grades yielded an average of 88.2% on the questions for this objective. This was the second highest, which is not surprising – logic was the first concept studied, and both of the midterms had significant questions on this topic. Thus, due to the increased exposure to the material, it is expected that the students would do well on it.

Objective 2: Proofs: Develop an understanding of how to read and construct valid mathematical arguments (proofs) and understand mathematical statements (theorems), including inductive proofs. Also, introduce and work with various problem solving strategies and techniques.
The final exam grades yielded an average of 81.7% on the questions for this objective. This shows solid understanding of the material. This value is lower than last semester, partly because more proofs (and more difficult proofs) were used this semester.

Objective 3: Counting: Introduce the basics of integer theory, combinatorics, and counting principles, including a brief introduction to discrete probability.

The final exam grades yielded an average of 72.7% on the questions for this objective. This was the lowest of the five objectives, yet it by itself is not significantly low. This objective included a lot of material that cannot be covered in depth in this course (such as discrete probability). While this is an area to focus on for the future, it is still rather positive that the lowest objective average was as high as 72.7%.

Objective 4: Structures: Introduce and work with important discrete data structures such as sets, relations, sequences, and discrete functions.
The final exam grades yielded an average of 82.7% on the questions for this objective. This shows solid understanding of the material.

Objective 5: Applications: Gain an understanding of some application areas of the material covered in the course.
The final exam grades yielded an average of 91.0% on the questions for this objective. This metric is a bit biased, however. The question was fairly easy, and there was only one question for which this statistic is measured. This objective was added for this semester, and was meant as a means of exposure – meaning that the students should be familiar with the applications of the material, so they can see how it applies to real world situations.

Assessment of Changes Made in the Course:

There were a number of aspects that I tried to improve upon from last semester, when I also taught the course.

The in-class presentations last semester did not seem to be as helpful as I had hoped. The students rated the various aspects of the course, and how helpful each one was. The in-class presentations received some of the lowest scores as to usefulness of the course materials. They were removed for this semester, which allowed for more room for additional course material.

The material taught in the course is difficult to easily see how it applies to real-world solutions. To this end, I put more focus on applications of the course content (and included it as one of the course objectives). A few additional topics were covered (such as logic gates) so the students could see better applications of the material being taught.

The homework assignments received some modification as well. This semester, more smaller assignments were given (22 total), rather than last semester, which had 11 larger assignments. This prevented them from only working on the assignments every week and a half, so they were more engaged in the material. Part way through the semester, the students expressed some discontent with the assignment grading, so that was modified as well (graded out of a 10 point scale instead of a 100 point scale, and standardizing what points were taken off for).

Other improvements included modifications to my teaching style, improving the slide sets, adding a bit more humor (to keep the class fun and keep them paying attention).

Other Issues:

1. Do you have concerns regarding the background of students coming into the course?

No. The students coming into the course are assumed to have no background in the material presented. This causes a lot of the initial material to be repeated for them. However, there is no obvious way to fix this – requiring a pre-requisite course (so the students all had the same background) would just move the problem to that course.

2. Are there other issues affecting student learning beyond what has been discussed elsewhere in this report? Include any other concerns you have about what students have or have not learned when they have completed the course.

   None.

3. If you know of changes being made or considered in the curriculum that might affect the course, briefly describe what these are and how the course might be affected.

Next semester, one of the CS 202 classes is going to be taught with a software verification focus. While it is a worthy idea, a significant amount of material in the course will have to be removed to make room for the additional material on software verification. My concern is that the students will not know the material for which the course is designed. Learning about software verification is useful, but it is not the purpose of this course. If both of these can be achieved, then that would be ideal. My concern is that the core course material will be sacrificed for tangential material.

4. List any other comments you think the Committee that monitors our degree programs should know about this course this semester.

   None.

Mapping of Course Objectives to BSCS Outcomes:

CS Degree Outcomes: Students who graduate with a BSCS will…	**Course Obj. 1**	**Course Obj. 2**	**Course Obj. 3**	**Course Obj. 4**	**Course Obj. 5**
(1: Math & DLD) Have demonstrated comprehension in relevant areas of mathematics (including calculus, discrete math, and probability), and in the area of logic design.	D	D	D	D	D
(2: Fundamentals) Have demonstrated comprehension in fundamental topics of computing, including the intellectual core of computing, software design and development, algorithms, computer organization and architecture, and software systems.	F	F	F	F	F
(3: Analysis & Evaluation) Have applied knowledge of areas of computing to analyze and evaluate algorithms, designs, implementations, systems, or other computing artifacts or work-products. Application of this knowledge includes the ability to design, conduct and evaluate the results of experiments and testing activity.		F		F	F
(4: Build Solutions) Have applied knowledge of areas of computing to create solutions to challenging problems, including specifying, designing, implementing and validating solutions for new problems.	X	X	X	X	X
(5: Research Awareness) Be aware of current research activity in computing through activities including reading papers, hearing research presentations, and successfully planning and completing an individual research project in computing or its application.					X
(6: Broadening) Have demonstrated comprehension of subjects in the humanities, social sciences, and the natural sciences in order to broaden a student's education beyond engineering and computing.					
(7: Social and Professional) Comprehend important social, ethical, and professional considerations related to computing practice and research, and be able to apply this knowledge when analyzing new situations.					
(8: Post-graduation) Be prepared to enter graduate programs in computing or related fields, and be prepared to begin a professional career in computing.	X	X	X	X	
(9: Life-long Learning) Have demonstrated a self-directed ability to acquire new knowledge in computing, including the ability to learn about new ideas and advances, techniques, tools, and languages, and to use them effectively; and to be motivated to engage in life-long learning.	X	X	X	X	X
(10: Teamwork) Have demonstrated the ability to work effectively in a development team.					
(11: Communication) Have demonstrated the ability to communicate effectively (orally and in writing) about technical issues.	X	X	X	X	X
(12: Professional development practices) Comprehend important issues related to the development of computer-based systems in a professional context using a well-defined process to guide development.					