Computer Information Sciences

Assessment

2006-08

Brooklyn College

Department of Computer and Information Science

Our Mission

The mission of the Department of Computer and Information Science of Brooklyn College is to develop and maintain high-quality research and educational programs in computer and information science. The department is committed to transmitting and expanding knowledge in both fundamental and applied areas and in the growing number of interdisciplinary fields that computing encompasses and impacts. The department is also committed to maintaining an awareness of the current and anticipated needs of the community that it serves. To achieve this mission, the department provides degree programs to part-time and full-time students as well as a wide range of research opportunities at the undergraduate, graduate and doctoral levels. These opportunities are provided within the framework of a collegial environment with equal opportunities for all faculty, staff and students.

Departmental programs include the following:

Educational programs that:

- Present fundamentals of computer and information science that foster disciplinary and intellectual maturity,
- Promote analytical and critical thinking,
- Emphasize knowledge that is relevant to technological innovation and business needs,
- Prepare students both for the job market and advanced studies, and
- Provide computer and information science courses for students who are in other programs at the College.

Research opportunities that:

- Foster leading edge research in the major areas of computer and information science, and
- Encourage students to participate in research relevant to industrial or academic careers.

Learning Goals and Objectives

Goal 1: **To Understand the Parts of a Computer and the Hardware - Software Interface.**

1.1 To understand different-base number notations and how numbers and characters are represented in a computer.
1.2 To understand the parts of a computer and how they work together to perform computation.
1.3 To understand the stored program concept, its strengths in performing computation and its limits.
1.4 To be able to understand the organization of a computer operating system.

Goal 2: **To Develop Computer Programming Skills.**

2.1 To effectively use programming constructs (variables, control structures, functions, and arrays)
2.2 To gain skill in at least two programming languages.
2.3 To learn at least two different programming paradigms.
2.4 To be able to transform algorithms into programs.
2.5 To understand the implementation and application of stacks, queues, lists, trees and graphs and their use in developing programs.
2.6 To understand the implementation of programming languages.

Goal 3: **To understand, develop, and analyze algorithms.**

3.1 To be able to develop algorithms.
3.2 To utilize appropriate algorithms for sorting and searching.
3.3 To be able to analyze the efficiency of algorithms and programs.
3.4 To be able to compare two algorithms and programs for relative efficiency in terms of both time and space.
3.5 To understand the limits of computation.
3.6 To understand recurrences and recursion and recognize problems that can be solved via recursion.

Goal 4: **To Gain Breadth in Several Areas of Computer Science.**

4.1 To be able to formulate formal logical propositions and perform simple proofs.
4.2 To gain familiarity with the history of computing.
4.3 To gain knowledge about areas such as artificial intelligence, multimedia computing, methods of simulation, computer networking, database organization, and system simulation.

Goal 5: **To Develop and Apply Appropriate Professional Skills and Practices**

5.1 To develop problem-solving skills.
5.2 To gain the knowledge and skills to educate themselves as computer science continues to evolve.
5.3 To utilize effective documentation techniques.

B.S. Program in Computer and Information Science

GOALS	OBJECTIVES	COURSES												
														CIS 88.1
		CIS 1.5	CIS 4.1	CIS 11	CIS 15	CIS 22	CIS 23	CIS 24	CIS 25	CIS 26	CIS 27	CIS 28	CIS 38	CIS 60.1
1 : To Understand the Parts of a Computer and the Hardware - Software Interface	1.1: To understand different-base number notations and how numbers and characters are represented in a computer	X	X								X	X		
	1.2: To understand the parts of a computer and how they work together to perform computation	X	X						X		X	X	X	
	1.3: To understand the stored program concept, its strengths in performing computation and its limits	X	X		X						X	X	X	
	1.4: To be able to understand the organization of a computer operating system								X			X		
2: To Develop Computer Programming Skills	2.1: To effectively use programming constructs (control structures, functions, and arrays)	X	X		X	X		X	X	X				X
	2.2: To gain skill in at least two programming languages	X	X		X	X		X		X				X
	2.3: To learn at least two different programming paradigms					X		X		X				
	2.4: To be able to transform algorithms into programs.	X	X		X	X				X				X
	2.5: To understand the implementation and application of stacks, queues, lists, trees and graphs and their use in developing programs					X	X			X				
	2.6: To understand the implementation of programming languages		X					X						
3: To be able to understand, develop and analyze algorithms	3.1: To be able to develop algorithms	X	X		X	X	X			X				
	3.2:To utilize appropriate algorithms for sorting and searching	X				X	X							
	3.3: To be able to analyze the efficiency of algorithms and programs.		X			X	X							
	3.4: To be able to compare two algorithms and programs for relative efficiency in terms of both time and space	X	X			X	X							
	3.5: To understand the limits of computation						X							X
	3.6: To understand recurrences and recursion and recognize problems that can be solved via recursion			X	X	X	X							
4: To Gain Breadth in Several Areas of Computer Science	4.1: To be able to formulate formal logical propositions and perform simple proofs			X	X	X	X							
	4.2: To gain familiarity with the history of computing	X	X					X	X		X	X	X	
	4.3: To gain knowledge about areas such as artificial intelligence, multimedia computing, methods of simulation, computer networking, database organization, and system simulation.	X												X
5: To Develop and Apply Appropriate Professional Skills and Practices	5.1: To develop problem-solving skills	X	X	X	X	X	X			X				X
	5.2: To gain the knowledge and skills to educate themselves as computer science continues to evolves	X	X	X	X	X	X	X	X	X	X	X	X	X
	5.3: To utilize effective documentation techniques	X	X		X	X				X				X

Assessment Cycle

Goal/Objective to be assessed	Year	Method	Results/ Data*	Student Work Samples
To understand the concept of a variable holding a value, how a variable is declared and how it can change	06-07	Answer to question scored using a rubric. Attached.	Attached.	On file in department.
All objectives	06-07	Survey of seniors. Attached.	Attached.	On file in department.
Learning Goals	06-07	Survey of faculty supervising senior projects.	Attached	On file in department.
To understand, develop and analyze algorithms/ To analyze the complexity and efficiency of an algorithm in terms of time.	07-08	Answer to question scored using a rubric. Attached.	Attached.	On file in department.
All objectives	07-08	Survey of seniors. Attached.	In progress.	On file in department.
Learning Goals	07-08	Survey of faculty supervising senior projects.	In progress.	On file in department.

*We will switch to four categories in the future: excellent, good, fair, and poor.

Departmental Process for Analyzing and Sharing Assessment Results

The assessment results are reviewed by the Undergraduate Curriculum Committee. At an annual meeting in the spring, the department discusses the assessment results and what strengths and weaknesses they demonstrate. These discussions are the basis for the implementation of changes designed to improve student learning.

Recent action: To improve learning in CIS 1.5, the department has configured a lab in which the classes will meet. An hour has been added to the meeting time and instructors will supervise lab sessions with their students.

CIS 1.5 Outcomes Assessment

CIS 1.5 Introduction to Programming Using C++

Objective:

Understand the concept of a variable holding a value, how a variable is declared and how it can change

Question:

1. For each of the following, do what is requested (if possible, each answer should be one statement only):

a. Declare a variable named **vara** to hold a person's monthly salary, showing dollars and cents, and initialize the variable to one cent less than five thousand dollars

b. Declare a variable named **varb** to hold the name of a book, and initialize the variable to *My Friend Flicka*

c. Declare a variable named **varc** to hold an array of five integers, and initialize the first three array elements to 7

d. Declare a variable named **vard** to hold a single character, and initialize the variable to the first letter of your last name

e. Declare a constant named **VARE** whose value is fixed throughout the program at 6.28

Assessment Rubric:

Good Enough

- Student can declare the variables in at least three parts of the question correctly
- Student can initialize those declarations correctly
- Student can at least start a declaration and initialization for the other two parts of the question

Less than Good Enough:

- Student is not able to give the correct declarations for the variables
- Student is not able to initialize the variables correctly

Better than Good Enough:

- Student can declare the variables in four or more parts of the question correctly
- Student can initialize those declarations correctly

Senior Survey

As part of an overall departmental assessment of our strengths and weaknesses, we ask that students in CIS 60.1 (normally a senior level course) answer the following questions. Even though you may be taking 60.1 prior to your senior year, we ask that you complete the survey anyway.

Please note that you are asked to include your name for control purposes only and that your responses will **in no way** affect the outcome of 60.1 (the responses will not even be tabulated until after the semester).

For each of the items below, please indicate how your computing education at Brooklyn College has prepared you to meet each of the following goals. You should select one of the following:

A= very well
B= well
C= fair
D= not well
E= poorly

How has your computing education at Brooklyn College prepared you to meet each of the following goals?

1. To develop problem-solving skills.

2. To develop skill in computer programming.

3. To gain skill in at least two programming languages.

4. To learn at least two different programming paradigms.

5. To understand different-base number notations and how numbers and characters are represented in a computer.

6. To gain familiarity with the history of computing.

7. To understand the parts of a computer and how they work together to perform computation.

8. To understand the stored program concept, its strengths in performing computation and its limits.

9. To be able to develop algorithms and transform the algorithms into programs.

10. To utilize appropriate algorithms for sorting and searching.

11. To be able to analyze the efficiency of algorithms and programs.

12. To be able to compare two algorithms and programs for relative efficiency in terms of both time and space.

13. To understand the limits of computation.

14. To understand recurrences and recursion and recognize problems that can be solved via recursion.

15. To be able to formulate formal logical propositions and perform simple proofs.

16. To understand the implementation and application of queues, lists, trees and graphs and their use in developing programs.

17. To understand the organization of a computer operating system.

18. To understand the implementation of programming languages.

19. To gain breadth in several areas of computer science such as artificial intelligence, multimedia computing, methods of simulation, computer networking, database organization, and system simulation.

20. To apply classroom skills by doing internships and research projects with professors and external institutions.

21. To gain the knowledge and skills to educate yourself as computer science continues to evolve.

22. To utilize effective documentation techniques.

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
BROOKLYN COLLEGE – SPRING 2007

SUPERVISOR SURVEY

TO THE SUPERVISOR:

As part of an overall departmental assessment of our strengths and weaknesses, we ask that supervisors of students in CIS 60.1 (normally a senior level course) answer the following questions regarding the preparation of the student(s) for this type of course.

Completion of the survey is optional and the anonymity of both the student and the supervisor are being preserved (there are no names on this form).

Please indicate how well you feel your student has been prepared in each of the following areas by selecting one of the following:

A= very well
B= satisfactory
C= unsatisfactory

Please indicate A, B or C for each of the following indicating your evaluation of the student's mastery of:

1. _____ Parts of a computer and the hardware-software interface.

2. _____ Computer programming skills.

3. _____ Understanding, development and analysis of algorithms.

4. _____ Breadth in several areas of computer science.

5. _____ Problem-solving and practical career skills.

Please send this completed survey (either separately or along with the supervisor evaluation) to:

Mailbox, CIS 60 & 80 Program
Department of Computer & Information Science
Brooklyn College
2900 Bedford Avenue
Brooklyn, N.Y. 11210

Outcomes Assessment Results CIS 2006-2008

CIS1.5 RESULTS '06-'07

better than good enough	63	48 %
good enough	26	20 %
less than good enough	42	32 %
not submitted	0	0 %
total	131	

RESULTS OF STUDENT SURVEY 06-'07

Question number	A	B	C	D	E
1	12	9	0	0	0
2	13	5	3	0	0
3	8	7	5	1	0
4	8	8	2	2	1
5	13	8	0	0	0
6	6	6	8	1	0
7	7	10	3	1	0
8	3	14	3	0	1
9	7	9	3	2	0
10	8	9	4	0	0
11	7	8	3	3	0
12	5	8	3	5	0
13	3	11	5	2	0
14	12	8	1	0	0
15	5	12	4	0	0
16	13	5	2	0	0
17	6	10	5	0	0
18	9	7	5	0	0
19	7	6	5	0	2
20	3	7	7	2	0
21	15	4	1	0	1
22	4	7	6	1	2

Results of Supervisor Survey '06-'07

Question number	A	B	C
1	3	2	
2	5		
3	3	2	
4	5		
5	5		

CIS23 results 07-'08

better than good enough	11	32 %
good enough	9	26 %
less than good enough	7	21 %
not submitted	7	21 %
total	34	

CIS22 results	07-'08	
Better Than Good Enough	9	26 %
Good Enough	14	41 %
Less Than Good Enough	11	32 %
Total	34	

CIS 23 Outcomes Assessment

CIS Departmental Goal: To be able to understand, develop and analyze algorithms.

Course Objective: Students will be able to analyze the complexity and efficiency of an algorithm in terms of time.

Let A=A[1],…,A[n] be an array of *n* distinct integers, in which each integer is in the range [1…n+1]. That is, *exactly* one integer out of {1,…,n+1} is missing from A.

The goal is to find the missing number as efficiently as possible, by minimizing the number of comparisons.

PART I of this problem assumes that the integers in A appear in an arbitrary order.
PART II assumes that A is given as a sorted array, i.e. A[1]<A[2]< … <A[n].

For each part, describe an efficient algorithm to find the missing integer. Justify the correctness of your algorithm.

Next, analyze the worst case time complexity of your algorithm, using *O*-notation. Explain your analysis.

Assessment Rubric:

Good Enough:

Any correct algorithm with a correct analysis will be considered good enough. Here are some possibilities:

Part I: Sort the array (using e.g. mergesort) and then perform a linear search. Time: O(n log n).

Part II: Compare the elements in the array to 1,2,… until the missing number is found. Time: O(n).

another option that is “good enough”:

Part I: Search for each 1<=i<=n. Time: $O(n^2)$.

Part II: Compare all neighboring elements to see whether they differ by more than 1. Time: O(n).

Less than Good Enough:

If either the algorithm does not correctly solve the problem, or the analysis is incorrect, the answer is not good enough.

Better than Good Enough:

Better than good enough is a linear time solution for Part I, and a logarithmic solution for II.

Part I: Declare a counter array of size n, initialize to all zeros. Iterate through A, incrementing location A[i] in the counter array. The index of the location that remains zero is the missing integer. Time: O(n).

Part II: Since up until the missing integer we have A[i]=i, and following we have A[i]=i+1, we can do a binary search to locate the missing integer. Time: O(log n).

CIS 22 Outcomes Assessment

CIS Departmental Goal: To be able to understand, develop and analyze algorithms.

Course Objective: Students will be able to analyze the complexity and efficiency of an algorithm in terms of time.

For each of the following sorting techniques, specify the order (e.g. O(n)) for the best, average and worst case. Also specify what type of input will result in the best and worst case running times. (If you don't think a best or worst case input sequence was mentioned or is inapplicable, say so.) Also specify whether the sort is in place, or whether it requires additional space, and if so, how much space is required. *Be sure to provide all this information for each of the sorts below*.

You may use a chart like the one below, but copy it to your booklet.

	Best case	Average case	Worst case	In-place/space requirement
Bubble sort				
Selection sort				
Insertion sort				
Quicksort				
Binary tree sort				
Heapsort				
Merge sort				

Assessment Rubric:

Good Enough: (all of the following)

- The student gives correct results for most of the boxes, although there might be a number (up to 10) of incorrect answers.
- Even if all the answers are not all correct, the answers show that the student can distinguish between which sorts are mostly O(n**2) and which are mostly O(nlogn).
- Student shows an understanding of the difference between best case, average case and worst case running times (i.e., even if the answers are not correct, the student recognizes that the worst case running time cannot be less than the average case, and that the average case cannot be less than the best case.)

Better Than Good Enough: In addition to the above)

Almost all of the answers are correct (less than 5 incorrect).

.

Less Than Good Enough: (any of the following)

- Many answers (more than 10) are incorrect.
- Student does not understand the difference between best case, average case and worst case running times (i.e., worst case running is less than average case, or average case is less than best case)
- Student clearly does not recognize the difference between an O(n**2) sort and and an O(nlogn) sort. For example, a student who enters O(nlogn) for *most* of the entries of Bubble Sort, or O(n**2) for *most* of the entries of HeapSort. (However, one or two incorrect answers, such as O(nlogn) for best case of Bubble Sort, can still fall in the category of “good enough”.)