UNIVERSITY OF NEBRASKA AT OMAHA
COURSE SYLLABUS/DESCRIPTION

Department and Course Number	CSCI/Math 4150/8156
Course Title	Graph Theory and Applications
Course Coordinator	Dr. Hesham H. Ali
Total Credits	3
Date of Last Revision	02/2015

1.0 Course Description:

1.1 Overview of content and purpose of the course (Catalog description).
The main objective of this course is to introduce graphs as a powerful modeling tool that can be used to solve many practical problems in various fields. To achieve this goal, the course introduces the main concepts of graph theory, graph representations and the basic classes of graphs. Many well-known graph problems and associated graph algorithms are also covered. At the end of this course, the student should be able to apply the abstract concepts of graph theory in modeling and solving non-trivial problems in different fields of study.

1.2 For whom course is intended.
Junior, senior and graduate students in mathematics, computer science, Bioinformatics, Information Assurance and Engineering majors and minors.

1.3 Prerequisites of the course (Courses).
CSCI/MATH 2030 (or 2230) and CIST 1400 or permission of instructor.

1.4 Prerequisites of the course (Topics).
Sets, functions, operations, matrices, basic probability and statistics, induction, and computer algorithms.

1.5 Unusual circumstances of the course.
None

2.0 Objectives:

2.1 Study in depth and become familiar with the basic concepts of graph theory as well as the main classes of graphs;
2.2 Deploy graphs as a modeling tool and recognize graphs as an important modeling technique in several applications;
2.3 Apply the abstract concepts of graph theory in solving practical problems in various application domains;
2.4 Conduct a reading project (research project for graduate students) to highlight the role of graph theory in modeling complex systems and solve practical problems.

3.0 Content and Organization:

The course covers the basic definitions, concepts and algorithms related to classical graph theoretic problems. The course also covers a number of applications in which graph modeling is known to be useful. The main topics covered in the course may be summarized as follows:

3.1 Introduction and basic definitions. (2 contact hours)
3.2 Graph representations and graph isomorphism. (1 contact hours)
3.3 Directed graphs and their applications. (3 contact hours)
3.4 Trees and their special properties and applications. (6 contact hours)
3.5 Connectivity, Euler tours and Hamiltonian cycles. (6 contact hours)
3.6 Coverings and matchings. (2 contact hours)
3.7 Cliques and independent sets. (2 contact hours)
3.8 Vertex colorings and edge colorings. (6 contact hours)
3.9 Planar graphs and networks. (6 contact hours)
3.10 General applications of graphs in various fields such as scheduling, bioinformatics, circuit design and communication networks (wireless networks). (6 contact hours)

4.0 Teaching Methodology:

4.1 Methods to be used.
The coverage will be covered primarily through lectures (incorporating Problem Based Learning techniques), in-class group activities (think, pair and share), reading assignments, homework assignments and student projects.

4.2 Student role in the course.
Students will be expected to attend classes, participate in class activities including group exercises, read assigned reading assignments, complete assigned homework and complete a reading project.

4.3 Contact hours.
Three (3) hours per week.

5.0 Evaluation:

5.1 Type of student projects that will be the basis for evaluating student performance, specifying distinction between undergraduate and graduate, if applicable. For Laboratory projects, specify the number of weeks spent on each project).
An individual reading project is required for each undergraduate student. An individual research project is required for graduate students and students who would like to apply the course as a part of an Honors program or the college Challenge program. The research project is expected to have a high degree of depth and coverage. Conducting a research project instead of a reading project is optional for undergraduate students and can be used to

improve the course grade. A written project report is required for all projects. Possible project ideas are:

Advanced Graph algorithms

Genetic algorithms for graph applications

Graph applications in Bioinformatics

Algorithms for transportation problems

Graph algorithms for networks/telecommunication problems

Graph theory based approaches in solving VLSI problems

Distributed graph algorithms

Algorithms in mobile environments

Scheduling algorithms

Hypergraphs

Perfect Graphs

Distributed graph algorithms

Application of graph theory in high performance computing

5.2 Basis for determining the final grade (Course requirements and grading standards) specifying distinction between undergraduate and graduate, if applicable.

A possible scheme for undergraduate students:

Component	Grading
Exams	36%
Quizzes and homework	30%
Individual reading project	07%
Final exam	27%

A possible scheme for graduate students:

Component	Grading
Exams	33%
Quizzes and homework	28%
Individual reading/research project	14%
Final exam	25%

5.3 Grading scale and criteria.

The following is the possible grading scale and criteria:

Points	*Grade*
97-100%	A+
93-96%	A
90-92%	A-
87-89%	B+
83-86%	B
80-82%	B-

77-79%	C+
73-76%	C
70-72%	C-
67-69%	D+
63-66%	D
60-62%	D-

6.0 Resource Material

6.1 Textbooks and/or other required readings used in course.

6.1.1 G. Agnarsson and R. Greenlaw, “Graph Theory: Modeling, Applications, and Algorithms,” Pearson Prentice Hall- 2007.

6.1.2 Hesham H. Ali and Naveed Sherwani, “Introduction to Graph Algorithms,” to be published by John Wiley & Sons, 2009.

6.2 Other suggested reading materials, if any.

Papers or texts related to student projects. The materials are expected to be different for each student.

6.3 Other sources of information.

6.3.1 GRPAHITE: A Software Application Tool for illustrating and analyzing fundamental graph algorithms

6.3.2 Course web site on blackboard.

6.4 Current bibliography of resource for student’s information.

6.4.1 Alfred Aho, John Ullman and Jeffery Hopecraft, “The Design and Analysis of Computer Algorithms,” Addison-Wesley, 1974.

6.4.2 J. A. Bondy and U. S. R. Murty, “Graph Theory,” Graduate Texts in Mathematics, Springer, 2008.

6.4.3 A. Brandstadt, V.B. Le and J.P. Spinard, “Graph Classes: A Survey,” SIAM, 1999.

6.4.4 Alan Dolan and Joan Aldous, “Networks and Algorithms,” John Wiley & Sons, 1993.

6.4.5 Michael Garey and David Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness,” W. H. Freeman and Company, 1979.

6.4.6 Ronald Gould, ``Graph Theory," Benjamin/Cummings, 1988.

6.4.7 Martin C. Golumbic, “Algorithmic Graph Theory and Perfect Graphs,” Academic Press, 1980.

6.4.8 Jonathan Gross and Jay Yellen, “Graph Theory and Its Applications,” CRC Press, 1999

6.4.9 Ellis Horowitz and Sartaj Sahni, “Fundamentals of Computer Algorithms,” Computer Science Press. 1984.

6.4.10 Fred S. Roberts, ``Graph Theory and Applications to Problems of Society," SIAM, 1987.
6.4.11 Gregory Rawlins, "Compared to What," Computer Science Press, 1992.
6.4.12 Robert E. Tarjan, "Data Structures and Network Algorithms," SIAM, 1983.
6.4.13 Robin J. Wilson and John J. Watkins, "Graphs: An Introductory Approach," John Wiley & Sons, 1990.
6.4.14 Douglas West, "Introduction to Graph Theory," second edition, Prentice-Hall, 2001.

7.0 (CS Program) Estimate Computer Science Accreditation Board (CSAB) Category Content (class time in hours):

CSAB Category	*Core*	*Advanced*
Data structures	3	2
Computer organization and architecture	1	0
Algorithms and software design	4	3
Concepts of programming languages	2	1

8.0 Oral and Written Communications:

Every student is required to submit at least __one___ written report (not including exams, tests, quizzes, or commented programs) to typically 12-18 pages and to make __0___ oral presentations of typically _____ minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

9.0 Social and Ethical Issues:

Applications of graph theory in social sciences are briefly introduced to motivate the use of graph techniques to model and solve problems in various fields of study. Basic definitions such as Balanced Graphs, Global Graphs and Six Degrees of Separation are also briefly covered.

10.0 Theoretical content:

The basic concepts of graph theory are covered with emphasis on the theoretic aspects of graphs in addition to the applied and problem solving aspect. The theoretic content represents the major component of the course:

10.1 Definitions and Terminology. (1 contact hour)
10.2 Graph isomorphism. (1 contact hour)
10.3 Directed graphs. (1 contact hours)
10.4 Trees and their special properties. (4 contact hours)
10.5 Connectivity, Euler tours and Hamiltonian cycles. (6 contact hours)

10.6 Coverings and matchings. (2 contact hours)
10.7 Cliques and independent sets. (2 contact hours)
10.8 Vertex colorings and edge colorings. (6 contact hours)
10.9 Planar graphs and networks. (6 contact hours)

11.0 Problem analysis:

Graphs are considering by many to be one of the most powerful tools for modeling and solving a large number of problems in diverse fields. Due to the ability to automate many graph related solutions, graph theory has become a key topic for computer scientists, mathematicians and engineers. This course addresses with a high degree of depth the issues of problem solving and efficient algorithm design in the domain of graphs. A good understanding of graph theory and graph algorithms should make it easier to get the most out of other mathematics, engineering and computer science courses and provides various, and in many cases better, ways to solve problems in various fields. At the end of the course, they will be able to apply the abstract concepts of graph theory in analyzing and modeling several practical problems with a particular focus on Biological and social networks.

12.0 Solution design:

The main objective of this course is to introduce graphs as a powerful modeling tool that can be used to solve many practical problems in various fields. A number of well-known graph problems and associated graph algorithms are covered. Once a problem is modeled by a graph, the problem domain is transformed to the domain of graphs in which the many well-researched graph algorithms can be utilized. At the end of this course, the student should be able to apply the abstract concepts of graph theory in modeling and solving non-trivial problems in different fields of study. Examples in how this process is conducted in Bioinformatics, wireless networks, chip design, transportation are covered in the class to give the students clear ideas of how graph theory is used to design solutions in various domains.

B. SUGGESTIONS FOR USE OF THE SYLLABUS FORMAT

I. <u>COURSE DESCRIPTION (1.0)</u>

The **course description** should give a general overview of the content and the purpose of the course (should be consistent with the Catalog descriptions), and a description of the kind of student who is expected to take the course. In addition, prerequisites for taking the course should be stated. Unusual circumstances of a course (1.4) might be an exceptionally small class size limitation or other restrictions or requirements for enrollment in the course.

II. <u>OBJECTIVES (2.0), CONTENT (3.0), TEACHING METHODOLOGY (4.0), AND TEXTBOOK(S) OR OTHER READING MATERIALS (6.0)</u>

The **objectives (2.0), content (3.0), teaching methodology (4.0), and textbook(s) or other reading material (6.0)** to be used in the course should be determined by the person most knowledgeable in the subject matter, i.e., you, the instructor. However, it is necessary to describe all of these in sufficient detail so that both the instructor (you may not always be the instructor) and the student can be sure of what the educational experience will be in the course. The attached sample syllabus may be a guide in this respect. (To eliminate the need for constant updating of 6.1, textbook(s), "or more recent text" can be added in parentheses after the current textbook used in the course).

Where the contact hours (4.3) are according to University standards (for example 3 hours contact time per week for a 3-credit-hour course), the contact hours need not be specified. They should be specified for practica, workshops, student teaching, counseling, or other courses involving teaching methods other than conventional classroom.

V. <u>STUDENT PROJECTS (5.1)</u>

Student projects (5.1), if any, should be described in sufficient detail so that the student knows whether the result of his\her work effort will be a written presentation, and/or performance of some specific kind. For dual level courses, i.e., 8--5 or 8--6, it is essential that the type of projects required of graduate students be clearly identified. The syllabus should describe those parts of the assignment upon which the student's grade will be based (5.2), for example: term papers, exams, oral presentations, class participation, etc. A percentage scale may be used, if desired, indicating the weight placed on each of these items in determining the final grade. For dual level courses, the evaluation must include some distinct and measurable item(s) of performance required of the graduate student that is not required of the undergraduate student. In addition, a grading scale should be provided to the students.

UNIVERSITY OF NEBRASKA AT OMAHA
Mapping of CS Program Outcomes vs. course objectives

Department and Course Number	CSCI/Math 4150/8156
Course Title	Graph Theory and Applications
Course Coordinator	Dr. Hesham H. Ali
Total Credits	3
Date of Last Revision	02/2015

Instructions: Paste or type the course objectives in the left-hand column. Indicate the relationship between course objective and program outcome by placing one of the two following marks in the appropriate cell:

S – Strong relationship
X – Contributing relationship

Course objective	**CS Program Outcomes**										
	(a) knowledge of discipline	**(b) analyze problem, define**	**(c) design and implement**	**(d) function on a team**	**(e) ethical issues**	**(f) communicate effectively**	**(g) analyze impact of computing**	**(h) continued professional**	**(i) Current techniques and tools**	**(j) apply foundations**	**(k) apply design and**
1. Study in depth and become familiar with the basic concepts of graph theory as well as the main classes of graphs	**S**						**S**	**X**		**X**	
2. Deploy graphs as a modeling tool and recognize graphs as an important modeling technique in several applications	**S**	**S**		**X**		**X**	**S**	**S**		**S**	
3. Apply the abstract concepts of graph theory in solving practical problems in various application domains	**S**	**S**	**X**			**X**	**S**	**S**	**S**	**S**	**X**
4. Conduct a reading project (research project for graduate students) to highlight the role of graph theory in modeling complex systems and solve practical problems.	**S**	**S**	**X**				**X**	**X**	**X**	**X**	**X**
5.											
6.											
7.											
8.											
9.											
10.											
11.											

CS Program Outcomes (2008)

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline;
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
(c) An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
(d) An ability to function effectively on teams to accomplish a common goal;
(e) An understanding of professional, ethical, legal, security, and social issues and responsibilities
(f) An ability to communicate effectively with a range of audiences
(g) An ability to analyze the local and global impact of computing on individuals, organizations and society
(h) Recognition of the need for, and an ability to engage in, continuing professional development
(i) An ability to use current techniques, skills, and tools necessary for computing practices
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
(k) An ability to apply design and development principles in the construction of software systems of varying complexity.