CS 4120 (1001) - Design and Analysis of Algorithms

BGSU Computer Science **Fall 2015 Syllabus**

Instructor: Dr. Robert Dyer
E-mail: **rdyer@bgsu.edu**
Office: HAYES 244
Phone: (419) 372–3469

Office Hours: MW 2:45pm – 4:30pm
TR 3:45pm – 5:00pm
OR by appointment (email me)

Class Meeting Time: Sec 1001: Tuesdays and Thursdays: 2:30pm - 3:45pm, HAYES 117

Textbook: "Introduction to Algorithms," by Cormen, Leiserson, Rivest, and Stein, MIT Press, 3rd Edition, 2009, ISBN-Hardcover: 978-0-262-03384-8, ISBN-Paperback: 978-0-262-53305-8.

Students Outcomes for the Course: After successfully completing CS 4120, students will be able to say:

- I can determine the complexity of an algorithm.
- I can explain and implement different types of algorithms (e.g., Divide-and-Conquer, Dynamic Programming, Greedy Algorithms).
- I can explain and implement different graph algorithms.
- I understand the implications of algorithm design in real-world applications.
- *I understand the classes of algorithms (P, NP, and NP- complete) and the role of polynomial-reduction in establishing NP-completeness.*

[Note: The outcome in *italic* is emphasized more for CS 5120, since it is more advanced; however, CS 4120 students are still expected to achieve all outcomes.]

Learning Outcomes for the BS CS Program Supported by this Course:

- Analyze a given problem, and identify and define the computing requirements appropriate to its solution; and
- Apply mathematical foundations, algorithmic principles, and computer science theory as appropriate in modeling and solving real-world problems.

[See: http://www.bgsu.edu/archived-catalog/fall-2014/colleges-and-programs/college-of-arts-and-sciences/computer-science.html]

Grading: The final grade will be composed of the following weights.
(The instructor reserves the right to make changes at any time.)

Assessments

Item	*Weight*
3 Exams	2 * 20% + 10% = 50%
Best 4 of 6 Homeworks[#]	4 * 6% = 24%
2 Projects	13% + 13% = 26%

Grading Scale

Range	*Grade*
[90–100]%	A
[80–90)%	B
[70–80)%	C
[60–70)%	D

[#]**All homework sets must be submitted**, and the *best four* will count toward the final grade.
Any missing or incomplete homework will be graded ***as is*** and count as one of the 4 graded homeworks.

Homework and Projects: Homework is essential to learning the materials, and you are expected to submit all homework assignments. The best four of six homework sets will be counted toward the final grade, each allowed one week for you to work on. Although the lowest two homework grades will be dropped, you are expected to put in reasonable effort toward all of them. To ensure that, if a homework is not turned in or

reflects substantially incomplete work, at the instructors discretion, that homework will count as one of the four graded homeworks. Two projects involving programming will be given during the semester, to be completed outside of class. You will be given three weeks to complete each project.

You are responsible for planning ahead to allow yourself enough time to complete all homework sets and projects by the deadlines. **Start your work early**. Homework and projects are due at the BEGINNING of the class on the due date. Absolutely **no late submissions** are allowed.

Exams: This course has 3 exams. While no exam is comprehensive per-se, they typically build on the material from the previous exam. All 3 exams are held in the normal classroom and will take an entire lecture period. The third exam is given the week before finals and the finals timeslot is used for presentations. The worst of the 3 exams will be graded as 10% of your overall grade and the others will be 20% each.

Course Objectives: This course teaches key algorithm development methodologies: greedy approach, divide and conquer, dynamic programming; important graph algorithms, several canonical problem types, and fundamentals of computational complexity.

Prerequisites: CS 3350 (Data Structures) and MATH 2220 (Discrete Math), or equivalent, including the basics of arrays, linked lists, stacks, queues, trees, and recursion.

Withdrawal Deadline: Friday, November 13, 2015. University policy states that after this date, anybody withdrawing from the course will have the grade automatically turn into an F.

Attendance: Students are expected to attend each class and be on time. I take attendance at the start of each lecture. I typically use good attendance as a factor when considering final grades. I reserve the right to penalize students up to 1% of their final grade, per absence, for more than 3 un-excused absences.

Class Notes/Handouts: Every student is responsible for taking notes in the class, collecting class handouts, and generally keeping up with the class, even if they must miss a class meeting for any reason. I do not post class notes, so if you miss a class please borrow notes from another student.

Canvas: The syllabus, schedule, and course policies will be available on Canvas. Your grades will also be available on Canvas throughout the semester.

Office Hours and Help: Please check your Canvas course site, Canvas messages, and your BGSU email **regularly**. *[You may have your Canvas messages forwarded to your BGSU/other email, and have your BGSU email forwarded to another favorite email address, if necessary, but do check it (multiple times) daily.]* I do forward my own Canvas messages to my BGSU email and check my BGSU email multiple times everyday (with rare exceptions). I will do my best to accommodate you ASAP, even if outside my posted office hours and without appointment. In general, if you need to see me in my office outside of my regular office hours, please make an appointment.

Academic honesty: All coursework for this class is expected to be YOUR OWN work. The MINIMUM penalty for copying someone's work (including current classmates, students from a previous offering of the course, or postings found on the web) or knowingly allowing someone to copy your work is a zero for the homework/project/exam/paper/presentation. The offense is also reported to the dean of your college. *Turnitin* and *Moss*, plagiarism detection tools, will be used in this course. I will follow the Department's policies and the University's code of academic conduct as defined in the *BGSU Student Handbook*. For specific details refer to:

1. *Department of Computer Science Academic Honesty Policy* (http://www.bgsu.edu/arts-and-sciences/computer-science/policies-for-current-students.html)
2. *BGSU Code of Academic Conduct* (http://www.bgsu.edu/content/dam/BGSU/student-handbook/documents/Academic-Code-of-Conduct-Chapter.pdf)
3. *The Academic Charter*, section B-I.G (http://www.bgsu.edu/content/dam/BGSU/faculty-senate/documents/academic-charter/B-I-G-Academic-Honesty-Policy.pdf)

Make-up policy: If you cannot take an exam as scheduled, you (or an authorized person, only in case you are unable to do so) must contact me ahead of time with the reason. Make-ups are considered typically for health emergencies only.

Disability Policy: In accordance with the University policy, students with disabilities must verify their eligibility through the Office of Disability Services for Students, 38 College Park Office Building, 419–372–8495 (http://www.bgsu.edu/disability-services.html). Contact me as soon as possible this semester to arrange any accommodations needed to assist with your success in this course.

Religious Holidays: It is the policy of the University to make every reasonable effort allowing students to observe their religious holidays without academic penalty. In such cases, it is the obligation of the student to provide the instructor with reasonable notice of the dates of religious holidays on which he or she will be absent. Absence from classes or examinations for religious reasons does not relieve the student of responsibility for completing required work missed. Following the necessary notification, the student should consult with the instructor to determine what appropriate alternative opportunity will be provided, allowing the student to fully complete his or her academic responsibilities (*The Academic Charter*, section B–I.F–4.b at: http://www.bgsu.edu/content/dam/BGSU/faculty-senate/documents/academic-charter/B-I-F-Classroom-Related-Responsibilities.pdf).

Tentative Course Schedule

Week	Day	Date	Topics / Important Dates	Assigned	Due
1	T	Aug 25	Introduction to course; Ch 1 – The Role of Algorithms in Computing		
	R	Aug 27	Ch 2 – Insertion sort; Ch 3 – Growth of Functions; Asymptotic notation		
2	T	Sep 1	Ch 4 – Methods for solving recurrences	**HW #1**	
	R	Sep 3	Ch 4 – Maximum subarray		
3	T	Sep 8	Ch 4 – Strassen's matrix multiplication		
	R	Sep 10	Ch 6 – Heapsort	**HW #2**	**HW #1**
4	T	Sep 15	Ch 7 – Quicksort		
	R	Sep 17	Ch 8 – Sorting in Linear Time	**Project #1**	**HW #2**
5	T	Sep 22	Catch up; Review for exam		
	R	Sep 24	**Exam #1** (ch. 1-4)		**Exam #1**
6	T	Sep 29	Ch 12 – Binary Search Trees		
	R	Oct 1	Ch 15 – Matrix chain multiplication		
7	T	Oct 6	Ch 15 – Longest common subsequence	**HW #3**	
	R	Oct 8	Ch 16 – Activity selection; Elements of the greedy strategy		**Project #1**
8	T	Oct 13	No Class – Fall Break		
	R	Oct 15	Data Encryption Algorithm (DEA)	**HW #4**	**HW #3**
9	T	Oct 20	Ch 16 – Huffman codes		
	R	Oct 22	Ch 22 – Representation of graphs; BFS; DFS;		**HW #4**
10	T	Oct 27	Review for exam		
	R	Oct 29	**Exam #2** (ch. 6-8, 12, 15)		**Exam #2**
11	T	Nov 3	Ch 22 – Topological sort; Strongly connected components	**Project #2**	
	R	Nov 5	Ch 13 – Red-Black Trees		
12	T	Nov 10	Ch 23 – Minimum Spanning Trees		
	R	Nov 12	Ch 24 – Single-Source Shortest Paths (SSSP)	**HW #5**	
13	T	Nov 17	Ch 24 – Dijkstra's algorithm		
	R	Nov 19	Ch 34 – NP-Completeness (Overview; Polynomial-time verification)		**HW #5**
14	T	Nov 24	Ch 34 – NP-Completeness continued	**HW #6**	**Project #2**
	R	Nov 26	No Class – Holiday		
15	T	Dec 1	Ch 34 – NP-completeness and reducibility		
	R	Dec 3	Ch 34 – NP-completeness proofs and NP-complete problems		**HW #6**
16	T	Dec 8	Catch up; Review for exam		
	R	Dec 10	**Exam #3** (ch. 16, 22-24, 34)		**Exam #3**
17	W	Dec 16	CS 5120 Presentations (CS 4120 students required to attend)		

NOTE: If there is a discrepancy between the due dates here and on actual assignments, the one on the assignment applies.