CS 210
ADVANCED DATA STRUCTURES & ALGORITHMS
FALL 2020

Department of Computer Science and Information Systems
Bradley University

BASIC COURSE INFORMATION:

- **Course Number:** CS 210
- **Course Name:** Advanced Data Structures and Algorithms
- **Section Number:** 01 & 02
- **Number of Credits:** 3 credits
- **Catalog Description:** Advanced topics in object-oriented programming with an emphasis on advanced data structures, algorithms, and software development.
- **Course Prerequisites:** A grade of C or better in both CS 102 and CS 140 or equivalents; MTH 120 or equivalent
- **Class Meeting Location, Days & Time:** BR 160, T & T, 9:00 a.m. - 10:15 a.m. (CS210-01) & BR 160, T & T, 3:00 p.m. - 4:15 p.m. (CS210-02)
- **Course Web Site:** http://hilltop.bradley.edu/~young/CS210FA20/

INSTRUCTOR:

- **Name:** Young Park
- **Title:** Professor of Computer Science
- **Office Location:** Bradley Hall 175
- **Office Hours:** T & T, 10:30 a.m. - 12:00 p.m. & 1:30 p.m. - 3:00 p.m. or by appointment. Walk-ins with short questions are encouraged anytime!
- **Contact Information:**
  - ***Telephone:*** (309) 677-2457
  - ***Fax:*** (309) 677-4504
  - ***E-mail:*** young@bradley.edu
  - ***Mailing address:*** Dept. of Computer Science and Information Systems, Bradley University, 1501 West Bradley Avenue, Peoria, Illinois 61625

COURSE OBJECTIVES, TOPICS & OUTCOMES:

- **Course Objectives:** The objective of this course is to introduce students to *advanced data structures, fundamental problems & algorithms, algorithmic analysis techniques*, *algorithm design paradigms* and *computational complexity of problems*.

- **Course Topics:** This course covers topics including

  – Advanced data structures such as *advanced linked lists (perfect skip list, randomized skip list and self-organizing list), graphs, advanced heaps (leftist heap and skew heap), advanced binary search trees (AVL tree and splay tree),* and *advanced multiway search trees (2-3 tree, 2-3-4 tree, Red-Black tree & B-tree).*

  – Algorithm complexity analysis techniques and algorithm design strategies such as *randomization, divide-and-conquer, dynamic programming, greedy approach, backtracking* and *branch-and-bound.*

  – Algorithms for fundamental problems including *searching, sorting* and *graph problems* and *computational complexity (lower bound)* and *intractability (NP & NP-completeness)* of fundamental problems including *searching, selection* and *sorting problems*.

- **Course Outcomes:** Upon completion of this course, the student should be able to

  ✓ *Understand* fundamental advanced data structures, to *describe* common applications for fundamental advanced data structures, to *design* new advanced data structures, and to *implement* major advanced data structures in C++.

  ✓ *Explain* mathematical concepts, techniques and notations used in describing the complexity of algorithms, to *analyze* worst-case and average-case complexity of both iterative and recursive algorithms, to *apply* a set of algorithm design strategies leading to the design of efficient algorithms to serve particular purposes, and to *implement* various different paradigm-based algorithms for fundamental computational problems in C++.

  ✓ *Show* computational complexity (lower bound) and intractability (NP-completeness) of fundamental computational problems.

COURSE MATERIALS:

- **Textbook:**
  – ***Lecture notes at http://bradley.bradley.edu/~young/CS210 (Required)***
  – ***Foundations of Algorithms, 5th ed., Neapolitan, Johns and Bartlett, 2014 (Required)***

- **Recommended Books on Data Structures & Algorithms (in C++):**
  – *Data Structures & Algorithm Analysis in C++*, 4th edition, M. Weiss, Addison Wesley Longman, 2014.
  – *C++ Data Structures: A Laboratory Course*, Third edition, S. Brandle, J. Geisler, J. Roberge & D. Whittington, Jones and Bartlett Publishers, 2009.

- *Introduction to Algorithms*, 3rd ed., Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, & Clifford Stein, The MIT Press, 2009.
- *Algorithms*, 4th edition, R. Sedgewick and K. Wayne, Addison Wesley, 2011.
- *The Algorithm Design Manuel*, 2nd edition, S. Skiena, Springer, 2012.

- **Recommended C++ Programming IDE & Pair Programming:**
  - *MS Visual Studio C++*
  - ***Pair Programming***
  - *Fun with Pair Programming*

COURSE GRADING:

This course uses the following absolute grading system. The course components consists of the following grading items:

- **Class preparation, attendance & participation, Review Quiz and Homework participation & contribution**
- **Homework Assignments –** Three *team* homework assignments
- **Midterm Tests –** Three midterm tests
- **Final Exam**

The numeric score (total: 1000 points) will be based on

- **Class preparation, attendance & participation, Review Quiz and Team Homework participation & contribution:** 100 points (= 45 points/CPAP + 45 points/RQ + 10 points/THPC)
- **Three Homework Assignments:** 300 points (= 3 x 100 points/ each HW)
- **Three Midterm Tests:** 300 points (= 3 x 100 points/ each Midterm Test)
- **Final Exam:** 300 points

The final letter grade will be determined from the total numeric score based on the following conversion rule:

LETTER GRADE	TOTAL NUMERIC SCORE RANGE
A	$900 < \text{raw-score} \leq 1000$
B	$800 < \text{raw-score} \leq 900$
C	$700 < \text{raw-score} \leq 800$
D	$600 < \text{raw-score} \leq 700$
F	$0 \leq \text{raw-score} \leq 600$

Course Policies:

- **Academic Honesty:** *Anyone found cheating on any graded assignment and examination will receive an F for this course and other further action will be taken.*
- **Attendance & Preparation:** *Lecture attendance is mandatory and students are expected to come well prepared for every class. Note taking is highly encouraged to help understand ideas more deeply.*
- **Assignment Submission:** ***All assignment must be handed in at the beginning of the class on the due date and in envelopes.***
- **Late Policy:** No late submission of assignment without a note from your doctor or your employer is accepted.
- **Makeup & Incomplete:** Makeup work and incompletes are only given in unusual circumstances, and only when work has been completely satisfactorily up to the point when the incomplete was requested.

Course Contents & Schedule:

(Tentative and subject to change)

Date	**Learning TOPICS**	(Before-Class Activity) **READINGS (Lecture Notes & Textbook)**	(Before-Class Activity) **Lecture VIDEO**	(In-Class Activity) **Lecture REVIEW & QUIZ**	(After-Class Activity) **HW**
8/27	Course Introduction				
9/1 **9/3**	LESSON 1: Algorithms: Efficiency & Analysis	**Algorithms: Efficiency and Analysis** **Chapter 1:** 1.1, 1.2, 1.3 (1.3.1 & 1.3.2 only), 1.4 (1.4.1 & 1.4.2 only), 1.5 **Appendix A:** A.1, A.5 **&** **Appendix B:** B.3	**Video1**	**Quiz1**	

9/8 **9/10**	LESSON 2: Advanced Linked Lists – Skip lists & Self-organizing lists LESSON 3: Graphs & Graph Problems	**AdvancedLinkedLists** **Graphs**	**Video2** **Video3**	**Quiz2** **Quiz3**	**HW#1**
9/15 **9/17**	LESSON 4: Divide-and-Conquer Paradigm	**Divide-and-Conquer** **Chapter 2:** 2.1, 2.2, 2.3, 2.4, 2.5, 2.8	**Video4**	**Quiz4**	
9/22 **9/24**	LESSON 5: Dynamic Programming Paradigm	**Dynamic Programming** **Chapter 3:** 3.1, 3.2, 3.3, 3.4, 3.6	**Video5**	**Quiz5**	
9/24	*In-Class Exercises & Review for Test#1*				
9/29	**Test #1**				
10/1 **10/6**	LESSON 6: Advanced Heaps – Leftist heaps & Skew heaps	**AdvancedHeaps**	**Video6**	**Quiz6**	**HW#2**
10/8 **10/13**	LESSON 7: Greedy Approach	**Greedy Approach**	**Video7**	**Quiz7**	

		Chapter 4: 4.1, 4.2, 4.3, 4.5 (4.5.1, 4.5.2 & 4.5.3 only)			
10/15 **10/20**	LESSON 8: Backtracking LESSON 9: Branch-and-Bound	**Backtracking** **Chapter 5:** 5.1, 5.2, 5.5, 5.7 **Branch-and-Bound** **Chapter 6:** 6.1	**Video8** **Video9**	**Quiz8** **Quiz9**	
10/22 **10/27**	LESSON 10: Computational Complexity: The Sorting Problem	**Computational Complexity: *Sorting*** **Chapter 7:** 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8 (7.8.1 & 7.8.2 only)	**Video10**	**Quiz10**	
10/27	*In-Class Exercises & Review for Test#2*				
10/29	**Test #2**				
11/3 **11/5**	LESSON 11: Advanced Binary Search Trees – AVL trees & Splay trees	**AdvancedBSTs**	**Video11**	**Quiz11**	**HW#3**
11/10 **11/12**	LESSON 12: The Selection Problem & Algorithms	***The Selection* Problem & Algorithms** **Chapter 8:** 8.5 (8.5.1, 8.5.2, 8.5.3 & 8.5.4 only)	**Video12**	**Quiz12**	

11/12 **11/17**	LESSON 13: Advanced Multi-way Search Trees – 2-3 trees, 2-3-4 trees RB Trees & B-trees LESSON 14: Computational Complexity: The Searching Problem	**AdvancedMSTs** **Computational Complexity: *Searching*** **Chapter 8:** 8.1 (8.1.1 only), 8.3	**Video13** **Video14**	**Quiz13** **Quiz14**	
11/19 12/1	LESSON 15: Intractability: P, NP & NP-Complete problems	**Intractability: P, NP & NP-Complete** **Chapter 9:** 9.1, 9.3, 9.4	**Video15**	**Quiz15**	
12/1	*In-Class Exercises & Review for Test#3*				
12/3	**Test #3**				
12/8	*Course Review for Final Exam*				
12/12 (SAT) **(9:00am - 11:00am)**	**Final Exam**				

*** End of CS210 Course Information ***