CSCI 2315.11 Data Structures
Summer 2015

by Charles E. Phillips, Jr.

1. General Information

1.1 **Instructor:** Dr. Charles E. Phillips, Jr.
Office: Bayou Building 3128-8 **Office Hours**: TBD, posted on Blackboard
Phone: x3837 **E-mail:** phillips@uhcl.edu
Class Time: Tues/Thurs, 9:00-11:50 am **Classroom**: Bayou Building 3607

1.2 **Teaching Assistant (TA):** TBD **TA Email**: TBD
TA Office Hours: TBD

1.3 UHCL Quality Enhancement Plan (QEP) Motto: Applied Critical Thinking (ACT) for Lifelong Learning and Adaptability

This course has been authorized by UHCL as an Applied Critical Thinking (ACT) Course, which means that in addition to learning about the specified course content, students will be engaged with some or all of the Elements of Thought and Universal Intellectual Standards of critical thinking. The objective of an ACT course is to develop the student's ability to become skilled at analysis and evaluation by applying a set of intellectual tools that may be effectively used across all disciplines (as well as to the student's personal life). Based on the Foundation for Critical Thinking model (http://www.criticalthinking.org/), critical thinking involves thinking for a *purpose*, asking *questions*, using *information*, applying *concepts*, drawing *inferences and conclusions*, identifying *assumptions*, anticipating *implications and consequences*, and recognizing *points of view*. The Universal Intellectual Standards that are applied to the Elements of Thought of critical thinking in order to develop Intellectual Traits include *clarity, accuracy, precision, relevance, depth, breadth, logic, significance,* and *fairness*.

Critical Thinking in Computer Science in General and Data Structures in Particular

Computer Science is the scientific and practical study of computation and its applications. Its scope is well agreed upon. For example, in Wikipedia, it is "the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information." A thorough understanding and precise specification of the problem domain through modeling, with a clear understanding of all underlying assumptions, is a prerequisite for effectively using computer science to construct a computer-based solution. All elements of thought of critical thinking are essential in every step of the elaboration and modeling of the problem, and design, implementation, and maintenance of a computer-based solution.

In particular, the principle of algorithm efficiency (Big-Oh) must be addressed or the processing power of the most modern computer can be overwhelmed. In the design and use of data structures to make computing algorithms/computing more efficient, critical thinking is integrated in the process. Data structure and application designers use principles of problem solving and algorithm analysis (Big-Oh) to thoroughly understand and specify the problem. Programming concepts of iteration, recursion, and Object Oriented Programming (OOP), along with the suitable data structure are

applied to infer and construct suitable logical design solutions. The implications and consequences of the design are realized through the effective uses of applications to store and retrieve data effectively.

Fundamental and Powerful Concepts (FPC) of the Course

In ACT vocabulary, fundamental and powerful concepts form the foundation that permeates and unites a course. In this course, these concepts are:

(1). The efficiency of an algorithm is the amount of work accomplished by the computer (measured in time or number of processes used) versus the amount of data/information being manipulated. This efficiency is evaluated as Big-Oh.
(2). Specific data structures combined with the appropriate programming techniques can improve computing-efficiency?

1.4 Other Course Information

Textbook

Mark Weiss. Data Structures & Problem Solving using Java, **4th Edition**. Addison Wesley. 2010. ISBN-10: 0321541405. ISBN-13: 978-0321541406.

Course Pre-requisites

CSCI 3133, CSCI 3134 or CSCI 1470, CSCI 1471. NOTE: JAVA is key

1.5 Course Description

Course Catalog Description

Course covers advanced programming techniques and data structures including arrays, linked lists, queues and stacks; abstract data types, recursion, searching and sorting, binary trees, hashing techniques, elementary algorithm design and analysis, and more.

Course Description

Data Structures is a programming-based course with the main purpose of educating students on the application of various data structures in order to develop more efficient computer algorithms to solve computer processing-intensive problems.

In order to meet this purpose, students must understand:
- A systematic approach to solving problems
- Java programming constructs (Reference Types, Programming with Objects, Recursion)
- How to evaluate performance (What is an Algorithm, Analyze/Calc. Performance, Big-Oh)
- Basic data structures and types (Strings, Arrays, Linked Lists)
- How to build more complex data structures (Stacks, Queues, Trees, Graphs, Tables)
- Implementations of data structures (Searching, Sorting, Paths, Insertions, Deletions)

This course will also focus on strengthening students' problem solving skills and critical thinking abilities. Critical thinking has an essential role in problem solving. As stated above, the student needs to think for a purpose (solving problems) by asking questions (to understand the problem), using information (Design a solution), applying concepts (Implementation of algorithm), drawing inferences and conclusions (Testing solution).

Course Goals. Upon successful completion of this course, students will be equipped with understanding how data structures can be utilized to enhance program performance and how data structures can be used to design elegant computer algorithms. Students will also be able to analyze computer algorithms to determine efficiency (Big-Oh), understand recursion, arrays, linked lists, stacks, queues, searching, sorting, graphs, hashing, binary trees, binary heaps, abstract data types and software development.

1.6. Student Learning Outcomes (SLO). To achieve the Course Goals, students must:

1. Understand basic software engineering concepts such as design, implement, and test a solution when given a set of requirements.
2. Recognize when and how to use the following data structures: arrays, linked lists, stacks, queues and binary trees.
3. Implement sequential searching, binary searching and hashing algorithms.
4. Apply various sorting algorithms including bubble, insertion, selection and quicksort.
5. Understand recursion and be able to give examples of its use.
6. Understand the complexity of algorithms (Big Oh notation).
7. Understand implications of unethical conduct.

The same set of SLOs using the vocabularies of Applied Critical Thinking (SLO ACT):

1. Problem Solving. Demonstrate with *clarity* the *purpose and concepts* of using a problem solving process to understand, design, implement, and test a solution when given a set of requirements. Note: The Design step of this process requires the student think critically and it is essential the student accurately develop their own design by analyzing and evaluating the problem for implementation and subsequent testing.
2. Data Structures. Clearly state the logical use of the following data structures: arrays, linked lists, stacks, queues, and binary trees; and the *relevant implications and consequences* of each.
3. Searching. Implement with accuracy; sequential searching, binary searching, best-path searching and hashing algorithms; and know the *purpose* (retrieving and deleting) for each approach.
4. Sorting and Storing. Apply with precision various sorting algorithms including: Bubble, Insertion, Shell, Merge, and Quick sorts; and know the *purpose* for each approach.
5. Recursion. Demonstrate the ability to interpret a problem and logically determine the relevance of recursive programming. Accurately identify the use of recursion and its significance.
6. Algorithm Efficiency (Big-Oh). *Accurately* determine the complexity of algorithms for the *problems (questions at issues).* Again, understanding the relationship between the amount of data and the amount of work done on that data is a fundamental and powerful concept.
7. Ethics. Understand perspectives (points of view) on unethical conduct and to apply all relevant viewpoints (fairness) in good faith.

The ACT SLOs to be assessed are SLO #1, SLO #2 and SLO #6.

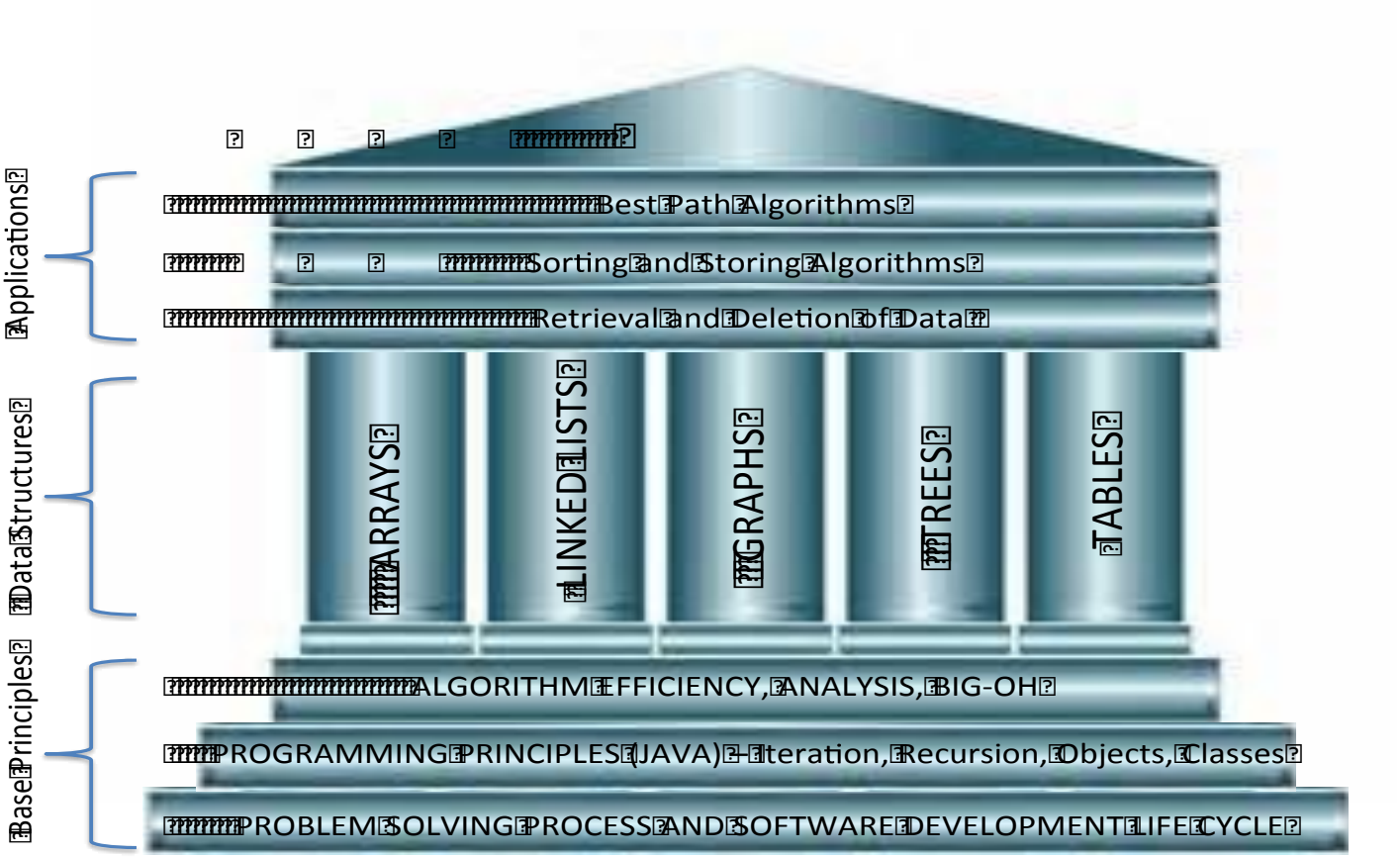
1.7. Central Questions

1. What is the efficiency (Big-Oh) of an algorithm (amount of work vs. the amount of data)?
2. What data structure is most efficient for a problem or computer algorithm?
3. How can data structures and programming techniques improve efficiency?

1.8. Course Format

- There will be a traditional lecture with weekly homework assignments and/or in class exercises.
- This is a face-to-face course conducted as lectures and presentations. The material will be posted on the course Blackboard. Students are expected to check Blackboard often.

1.9. Structure of Course Concepts



2. Course Policies and Guidelines

2.1 General Policies

1. Blackboard. The Blackboard site will be the official site for this course. All submissions and deliverables of assignments are due on date given on Blackboard unless otherwise specified.

2. Classroom conduct:
   - You are encouraged to ask a lot of questions in the classroom. Active participation is an essential in any kind of learning.
   - Be polite to your classmates. Please do not chat with your neighbors during classes.
   - Mobile phones and pagers should be turned off during classes.
   - If you must multi-task, keep it quiet.

3. EMAIL.
   - Must use official UHCL-mail only.
   - Please note course (CSCI 2315) in Subject Line.
   - Check your mail at least once per day.

- Be respectful in email correspondence.

4. Assignments.
   - Assignments are due according to the date posted on Blackboard.
   - No exceptions unless approved by instructor.
   - Late assignments are accepted with a penalty of 25% deduction per weekday after the due date for only two days. No late assignment will be accepted over two days after the due date.
   - Make sure that you follow the submission guidelines for assignments on Blackboard.
   - For consistency, if you have a dispute in homework or project grading, discuss with TA first.
   - Respect your TA, they are here to support you.

5. Examinations. No make-up exams except in verified emergencies with immediate notification.

6. Others.
   - No incomplete grade or administrative withdrawal under nearly all situations.
   - UHCL Information about withdrawals, appeals, GPA, repeated courses, the 6 drop rule, etc. can be found in the general program requirement section: http://www.uhcl.edu/XDR/Render/catalog/archives/125/06/%23A0110.
   - Students with special needs and disability should contact the instructor as soon as possible and contact Disability Services Office at 281-283-2627 website: www.uhcl.edu/disability

2.2 Attendance

- Students are expected to attend class regularly and actively participate in classroom discussions.
- Class attendance is expected and part of participation grade.
- It is the student's responsibility to get the material discussed, announcements, handouts, or anything conducted during a missed class meeting.
- Participation in discussions with/from students is highly encouraged.

2.3 Academic Honesty

Penalty on cheating will be extremely severe. Standard academic honesty procedure will be strictly followed, so use your best judgment. If you are not sure about certain activities, consult the instructor.

See: http://prtl.uhcl.edu/portal/page/portal/PRV/FORMS_POLICY_PROCEDURES/STUDENT_POLICIES/Academic_Honesty_Policy

The Honesty Code is the university community's standard of honesty and is endorsed by all members of the University of Houston-Clear Lake academic community. It is an essential element of the University's academic credibility. *It states: I will be honest in all my academic activities and will not tolerate dishonesty.*
Academic honesty is integral to university education. Students are advised to thoroughly understand UHCL academic honesty policy.

2.4 Academic Adjustment Policy

The University of Houston System complies with Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act of 1990, pertaining to the provision of reasonable academic adjustments/auxiliary aids for students with a disability. In accordance with Section 504 and ADA guidelines, each University within the System strives to provide reasonable academic adjustments/auxiliary aids to students who request and require them. If you believe that you have a disability requiring an academic adjustments/auxiliary aid, please contact your University's student disability services center.

2.5 Assessment for Accreditation

The School of Science and Computer Engineering may use assessment tools in this course and other courses for curriculum evaluation. Educational assessment is defined as the systematic collection, interpretation, and use of information about student characteristics, educational environments, learning outcomes, and client satisfaction to improve program effectiveness, student performance, and professional success. This assessment will be related to the learning objectives for each course and individual student performance will be disaggregated relative to these objectives. This disaggregated analysis will not impact student grades, but will provide faculty with detailed information that will be used to improve courses, curriculum, and student performance.

3. Grading Policy

3.1. Grades. Grades will be assigned based solely on homework (Home work and In Class Exercises (ICE)), examination (Midterm, Final, Quizzes) scores and Participation (Attendance and timely Assignment Submission). No other factors will be considered. I grade to the same standards for all Students. The following are not reasons to reconsider Student grades:

- Expected a higher grade
- Good improvement during the semester
- Have put in extra efforts
- Need to avoid probation
- Financial needs
- Loss of scholarship
- Loss of job opportunity
- Loss of practical training opportunity
- Need to graduate
- Company relocation
- Immigration status needs
- Family needs
- Sickness during the semester
- and others.

Additionally, There will also be no 'special project' that you can work on to improve your grades after the final examination. Anything I offer to one student, I will be offered to the entire class during the semester.

3.2. Grading and Evaluation

Participation	5%	(100points, Attendance, Timely Work Submission)
Assignments, ICE*	20%	(400 points; ~8@50 points)
Quizzes	20%	(400 points; ~4@100 points)
Midterm Exam	25%	(500 points)
Final exam	30%	(600 points)

* ICE – In Class Exercises

3.3. Grading Scale

93+ = A;	80-82.9 = B-;	67-69.9 = D+;
90-92.9 = A-;	77-79.9 = C+;	63-66.9 = D;
87-89.9 = B+;	73-76.9 = C;	60-62.9 = D-;
83-86.9 = B;	70-72.9 = C-;	< 60 = F

4. Course Schedule (Note: Schedule subject to minor change)

Week	Dates	Lesson	Reading
1	9 June 15	Introduction to software development, The Life Cycle, and Problem Solving	Chapter 1,2
2	11 June 15	Review Java Programming (Primitive Java, Reference Types, Objects, Classes)	Chapters 1,2,3
3	16 June 15	Algorithm Analysis (What's an Algorithm, Big O, Searching)	Chapters 5
4	18 June 15	Recursion	Chapter 7
5	23 June 15	Sorting	Chapters 6, 8
6	25 June 15	Arrays, ArrayLists, and Linked Lists	Chapter 17
7	30 Jun 15	Stacks and Queues	Chapter 16
8	2 July 15	**No Class**	
9	7 July 15	**Midterm Exam**	All to Date
10	9 July 15	Graphs and Paths	Chapter 14
11	14 July 15	Trees	Chapter 18
12	16 July 15	Binary Search Trees	Chapter 19
13	21 July 15	Binary Heap	Chapter 21
14	23 July 15	Hashing	Chapter 20
15	28 July 15	Computers and Professional Ethics / Final Exam Focus	
16	30 July 15	**Final Exam (9:00-1150am, Thursday, 30 July 2015)**	Comprehensive
Grades	TBD	Official Grades Available	

4.1 Important Course Dates:

Midterm:	7 July 2015
Final:	30 July 2015

4.2 Important School Dates: (You must confirm on the University Website)

8 June 2015 .....First Class Day for Summer15 (8-week Session)
16 June 2015 .....Census Date
20 June 2015 ……..Last Day to Drop/Withdraw
1 August 2015…….....Last Class Day Summer 15 (8-week Session)

5. Applied Critical Thinking (ACT)

5.1 Vocabulary of Critical Thinking

We use the vocabulary of critical thinking described by Drs. Richard Paul and Linda Elder, including the eight elements of thought and nine universal intellectual standards:

Eight elements of Thought of Critical Thinking:

1. Purpose
2. Question at Issue
3. Information
4. Interpretation and Inference
5. Concepts
6. Assumptions
7. Implications and Consequences
8. Point of View

Nine Universal Intellectual Standards for Critical Thinking:

1. Clarity
2. Accuracy
3. Precision
4. Relevance
5. Depth
6. Breadth
7. Logic
8. Significance
9. Fairnes

For more details, see:
[1] Paul, R. and Linda Elder, L., The Miniature Guide to Critical Thinking Concepts and Tools (Thinker's Guide), 7th edition, Foundation for Critical Thinking, 2014.
[2] Paul, R. and Linda Elder, L., Critical Thinking: Tools for Taking Charge of Your Learning and Your Life, 3rd Edition, Prentice Hall, 2011.

5.2 Critical Thinking Process (CTP)

According to the ACT vocabulary we used, there are four major aspects of the Applied Critical Thinking Process, termed as the 4 C's: *curiosity, connections, creativity and communication*. In this course, the C in the student learning objectives is *connections*:

- Making *connections* to a particular issue or problem: students will use established academic and industrial methodology to model a problem, design a database solution, and manipulate the data.

5.3 Critical Thinking Activities and Assessment

Critical thinking activities are integrated in the course. Lectures will include examples to highlight CT elements and standards, and their applications. Homework, In Class Exercises and Quizzes contain ACT components. Additionally, both the midterm and final examinations include ACT-oriented questions.

In particular, the data of two assessment activities (AA) on ACT will be collected to assess how well critical thinking is incorporated into the course. These assessments will be used as input to the UHCL Critical Thinking database for internal assessment of Critical Thinking, and will *not* affect your grade of the course.

1. Homework2 on Algorithms (HW2_ALG): Students will analyze a series of code segments in order to determine the Big-Oh (efficiency) of the algorithm and then implement the different algorithms in order to measure the run-times (amount of work by computer) for different amounts of data. This homework requires the students to have a *clear conceptual* understanding of the *theory* of algorithm analysis to determine the efficiency of an algorithm (SLO#6) and that the student can interpret the problem clearly and accurately in its application (SLO #1).
2. Quiz3 on Graphs_Paths_Trees (Q3_GPT): Students will have to address the question at issue and determine the correct data structure (stack, queue, and priority queue) to use in order to precisely and accurately calculate the correct outcomes (SLO #2).

This is related to Student Learning Outcomes (SLO) and Fundamental and Powerful Concepts (FPC):

ACT Assessment Activity	SLO	FPC
HW2_ALG	6, 1	1,2
Q3_GPT	2	1,2

The assessment criteria for the AA:

ACT Activity	Assessment Outcomes

	Unacceptable	Acceptable	Excellent
AA-ALG	[0%,74%]	[75%,89%]	[90%-100%]
AA-GPT	[0%,74%]	[75%,89%]	[90%-100%]

The course assesses connections out of the four C's.
Overall, if 70% or above of students are evaluated to be acceptable or excellent in each activity, as well as the average of all activities, the outcomes will be deemed acceptable. Overall, the instructor will evaluate the ACT content, activities, and assessment of the course and make necessary adjustment.