Engineering Division Course Syllabus

Course Title **Data Structures and Algorithms**

Course Number **ENGR-UH 3510**

Course Description: This course introduces students to the concepts of data structures and algorithms. A low-level programing language is used to illustrate the course topics, which may be C/C++ or a similar programming language. The course covers topics data structures including lists, stacks, queues, trees and graphs, and algorithms analysis.

Credits 4

Prerequisite Courses Recommended: ENGR-AD-202 Computer System Programming

Instructor Information

Instructor(s) Prof. Yi Fang (Instructor)

Contact Information: C1 Office: 156
Email: yfang@nyu.edu
Phone: 056-3190302

Intended Learning Outcomes: Upon successful completion of this course, students will be able to:

- Describe the fundamental concepts of data structures and algorithms such as behaviors of basic data structures, time complexity, functions, arrays, pointers, reference and class data members and function member [(b.3)];
- Analyze and solve the basic data sorting and searching problem using programming language [(b.3)];
- Apply knowledge of data structures and algorithms to analyze and model engineering problems to meet desired needs [(b.3),(c.2),(d.1)]
- Identify the bugs in source code of algorithm design and develop trouble-shooting skills [(e.3),(d.1)];
- Be able to understand medium-scale algorithms and find efficient ways to implement them to solve engineering problems [(e.3),(d.1)].

Course Materials

Textbook: Data Structures and Algorithm Analysis in C++, 4th Edition, Mark A. Weiss

Teaching and Learning Methodologies: Students are expected to arrive at class with an understanding of the basic definitions, concepts, and applications of relevant topics. Class time will be devoted to in-depth discussions of course topics. Lab project assignments are dedicated for reinforcing course topics via solving representative problem sets and holding class discussions.

Assignments and Grades: Grades will be based on six computer programming assignments, three midterm exam (paper and computer based) and one comprehensive Final exam (paper and computer based). The distribution of grades is subject to some revision at the discretion of the instructor. Typical weighting values are given below:

Projects	5*6%
Mid-term Exam	15*3%
Final Exam	25*1%
Total	100%

Course Schedule: A typical schedule for course topics, projects, and exam dates is given in the table below (The actual schedule might be slightly different from this).

Week	Lecture Topics	Project	Exam Schedule
1	▪ Introduction ▪ Algorithm analysis		
2	▪ Lists ▪ Stacks ▪ Queues	Project 1	
3	▪ Lists ▪ Stacks ▪ Queues	Project 1	
4	▪ Trees ▪ Ordered trees ▪ Search trees	Project 2 & 3	Midterm
5	▪ Trees ▪ Ordered trees ▪ Search trees	Project 2 & 3	
6	▪ Trees ▪ Ordered trees ▪ Search trees	Project 2 & 3	
7	▪ Sorting Algorithm • Insertion sort • Bubble sort • Heap sort • Merge sort • Quicksort	Project 4	
8	▪ Sorting Algorithm • Insertion sort • Bubble sort • Heap sort • Merge sort • Quicksort		Midterm
9	▪ Hash functions ▪ Hash tables	Project 5	
10	▪ Hash functions ▪ Hash tables		
11	▪ Graph algorithms	Project 6	
12	▪ Graph algorithms		Midterm
13	• Algorithm design		
14	▪ Algorithm design		Final Exam

Course Structure: 14 weeks course, 2 lectures per week (75 minutes each) + 155 minutes lab per week, 2 hours midterm, and 2 hours final exam.

Relationship to Outcomes:

	Shared Engineering Outcomes					Program Specific Criteria				
	(1)	(2)	(3)	(4)	(5)	CivE	CmpE	ElecE	MechE	GenE
Lectures		x	x	x	x		X			x
Labs		x	x	x	x		x			x

	Student Learning Outcomes												
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)
Lectures		x	x	x	x								
Labs		x	x	x	x								

Shared Engineering Outcomes:

(1) Apply techniques in the practice of leadership and innovation [(l), (m)];
(2) Identify social, economic, ethical and other factors that shape engineering solutions and incorporate them in conjunction with engineering principles in problem solving and designing systems, components, or processes to meet desired needs within realistic constraints [(a), (c), (e), (f), (h)];
(3) Recognize and respond respectfully to cultural concerns and differences when solving problems both physical and ethical [(a), (e), (f), (h)];
(4) Exhibit guidance and organizational effectiveness in multidisciplinary teams as a participant and a leader [(d), (l)];
(5) Demonstrate competence in writing and speaking effectively, and in communicating significant technical information in a clear and concise manner [(g)].

Program Specific Criteria:

- CivE: Civil Engineering graduates will be able to work professionally in four of the technical areas of the civil engineering discipline (structural, geotechnical, transportation, and environmental), design systems, components, and processes in more than one civil engineering context, and apply the principles of project management.
- CompE: Computer Engineering graduates will be able to analyze and design complex computing and network devices and systems containing hardware and software components.
- ElecE: Electrical Engineering graduates will be able to analyze and design complex electrical, electronic, and communication devices and systems.
- MechE: Mechanical Engineering graduates will be able to analyze and design systems, components, and processes, and work professionally in both thermal and mechanical systems areas.
- GenE: General Engineering graduates will be able to analyze and design devices and systems in an interdisciplinary engineering area related to: Biomedical and Health Systems; Information, Communication, and Electronic Systems; or Urban Systems.

Student Learning Outcomes:

(a) an ability to apply knowledge of mathematics, science, and engineering;

(b) an ability to design and conduct experiments, as well as to analyze and interpret data;

(c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability;

(d) an ability to function on multidisciplinary teams;

(e) an ability to identify, formulate, and solve engineering problems;

(f) an understanding of professional and ethical responsibility;

(g) an ability to communicate effectively;

(h) the broad education necessary to understand the impact of engineering solutions in a global, economic, environmental, and societal context;

(i) a recognition of the need for, and an ability to engage in life-long learning;

(j) a knowledge of contemporary issues;

(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice;

(l) an ability to apply leadership skills including risk and project management and decision making;

(m) an ability to apply innovation skills.

Assessment Plan for ENGR-AD 202: Computer System Programming

Performance Indicators	Assessment Tools		
	Project	**Midterm Test**	**Final Exam**
(b.3) Analyze and interpret data related to engineering experiments. *Analyze and interpret data from various engineering applications, for example, 2D image data that helps student using computer programming model real problems*	✓		✓
(c.2) Design a component to meet desired needs within realistic constraints. *Apply C++ class object oriented programming principles and procedure to take a list of specifications and design a class and algorithm while meeting memory and time complexity requirements*	✓		✓
(d.1) Fulfill individual duties. *Deliver lab projects by the deadline and finish the exams on time.*	✓	✓	✓
(e.3) Demonstrate the ability to solve engineering problems using appropriate methods. *Demonstrate ability to implement an algorithm to find a complete solutions of a well-defined engineering problems, for example, comparing similarity between a pair of images*	✓		✓
CompE: ***Criteria satisfied by outcomes above and degree curricular requirements.***			
GenE: ***Criteria satisfied by outcomes above and degree curricular requirements.***			