CPE 112 Programming with Data Structures

2/2023

Instructor:	Assoc. Prof. Dr. Natasha Dejdumrong	**Email:**	natasha.dej@mail.kmutt.ac.th
TA:	Marnie Gorgonia	**Email:**	marnie.gorg@kmutt.ac.th
	Kittipong Tapyou		kittipong.tapy@kmutt.ac.th
	Wongsatorn Sungsilpawech		wongsatorn.sung@kmutt.ac.th
	Apichat Aimimpak		apichat.aimi@kmutt.ac.th
Class Times	(A) Thu 8:30 - 10:30	**Lab:**	(A, B) Thu 13:30 - 15:30
	(B) Thu 10:30 - 12:30		

COURSE CATALOG:

Computer programming course with an emphasis on dynamic data structures such as dynamic arrays, linked lists, trees, graphs and hash tables. Creation of general, reusable modules and their use in multi-module software systems. Weekly lab sessions will focus on applications of the concepts covered in lectures.

COURSE DESCRIPTION: This course introduces students to the most important categories of data structures (strategies for organizing information in a computer program). In conjunction with the study of these data structures, we will consider the algorithms (methods for solving particular kinds of problems) used to work with them. By the end of the course, each student should be able to analyze a problem and determine appropriate data structures and algorithms to use for solving the problem.

Although the basic concepts of data structures and algorithms apply across programming languages, this class will build on the students' knowledge from CPE100 by using the C language for examples, demonstrations, lab exercises and projects. We will continue to emphasize the development of good programming skills, such as using design notations, following coding standards, and refactoring to produce modular and maintainable code.

Although there will be some lectures because the class is so large, the course will use active learning techniques wherever possible, including experimenting with demo programs, working on lab exercises, and designing and building a moderately complex term project.

MAIN REFRERENCES:

- Thareja, Reema. *Data structures using C.* Oxford University Press, Inc., 2011.

PROGRAM OUTCOMES:

An ability to apply knowledge of mathematics, science, and engineering

COURSE LEARNING OUTCOMES:

1. Identify and explain concepts of linear and non-linear data structures.
2. Analyze problems and implement in linear and non-linear data structures.

ULTIMATE LEARNING OUTCOMES:

M1: Explain properties and applications of linear data structures.

M2: Explain properties and applications of non-linear data structures.

M3: Select and implement appropriate data structures and associated algorithms to efficiently solve programming problems.

RUBRICS:

Criteria	**Performance descriptors**		
	Level 1: Below Expectations	**Level 2: Meets Expectations**	**Level 3: Exceeds Expectations**
Identify	Does not realize when major components of the data are missing and has no coherent strategies for using data structures	Has a vision of the whole data and has some stratefies for using data strucutes	Demonstrates understanding of how various structures of the data relate to each other and the whole
Implement	Does not see the connection between theory and practical data structures	Connects theoretical concepts to practical data structures when prompted and is beinging to integrate previous knowledge and new information	Implements structures for data and can relate theoretical concepts to practical data structures
Application	Does not understand the applciation of data structures in solving real world problems	Shows nearly complete understanding of applications of data structures in solving real world problems	Applies concepts of data structures to solve real world problems

GRADE POLICY:

Final grades are based on performance indicated by student in-class exercises, lab reports, homework assignments, quizzes, projects, module exam grades. The final grade will be calculated according to the following weights:

- **Grade Components (Tentative)**:
  - Module 1 .... 30%
    - Labs
    - Quizzes
    - Portfolio
    - Exam 1
  - Module 2 .... 40%
    - Labs
    - Quizzes
    - Portfolio
    - Exam 2
  - Module 3 .... 30%
    - Project
    - Application Review
    - Portfolio

- **Examinations**:

  There are one module 1 and one module 2. Both examinations will cover course materials from the beginning until the examinations. However, the focused materials for the final examination will be those covered since the midterm examination.

  There will be no make-up examination for either midterm or final examinations without prior consent from the instructor. If a student has a legitimate and verifiable reason, a separate comprehensive examination will be allowed.

- **Test, Labs and Assignments**:

  There will be no tests throughout the course. The course includes 9 weekly lab assignments. Students are allowed one week to complete each assignment.

- **Attendance**:

  Students are required to attend at least 80% of the lectures. Those students who fail to meet this requirement will be reported disqualified for sitting in the final examination. Active participation in class discussion is also expected.

OTHER COURSE POLICY:

The lectures will stress most of the important issues in the textbook. Students are responsible for all materials covered in the class and assigned reading. Lectures may go beyond the scope of the textbook for certain topics. Therefore, it is important for students to attend the class.

Students are responsible for all announcements and changes made in class.

To be successful in this course, students should be able to

- Understand the concepts presented in the lectures and textbook readings,
- Recognize these concepts both in definition and example forms, and
- Apply these concepts in assignments.

COURSE SCHEDULE:

The following is a very tentative course schedule. The instructor may revise parts of the outline to conform to the background, knowledge, and interests of the students.

Week	Topics	Hrs	Chapter	Instructor
[1] 18/01	Introduction to C Programming and Data Structure	2	1, 2	TA
	Lab 1 Sets	2		
[2] 25/01	Array, String, Structure, and Union	2	3,4,5	Dr. Natasha
	Lab 2 Arrays	2		
[3] 01/02	Linked List	2	6	Dr. Natasha
	Lab 3 Linked List	2		

[4] 08/02	Stacks and Queues	2	7, 8	Dr. Natasha
	Lab 4 Stacks	2		
[5] 15/02	Active Learning: Linear Data Structures	2	6, 7, 8	Dr. Natasha
	Lab 5 Queue	2	7, 8	
[Exam 1]	Programming Exam 1 (19/02 - 23/02)			
[6] 29/02	Trees 1	2	9	Dr. Natasha
	Lab 6 Tree 1	2		
[7] 07/03	Trees 2 and Heap	2	10, 11	Dr. Natasha
	Lab 7 BST and AVL Tree	2		
[8] 14/03	Graph 1	2	13	Dr. Natasha
	Lab 8 Graph 1	2		
[9] 21/03	Graph 2	2	13	Dr. Natasha
	Lab 9 Graph 2	2		
[10] 28/03	No Class			
[Exam 2]	Paper-Based Exam & Programming Exam 2 (28/03 - 05/04)			
[Break]	Holidays (08/04 - 16/04)			
[11] 18/04	Sorting Algorithms 1	2		Dr. Natasha
	Project Initialization			
[12] 25/04	Sorting Algorithms 2	2		Dr. Natasha
	3 Sorting Programming Exam 1	2		TAs
[13] 02/05	2 Sorting Programming Exam 2	2		TAs
[14] 09/05	Project Presentation	2		Dr. Natasha & TAs
[Exam 3]	Final Exam (20/05 - 30/05)			