

Data Structures and Practice

Course Name	Course type (credit/hours)		Required course(4/5)		Course code	F031
	Target students Division/major/grade		Software and Computer Engineering/Sophomore		Opening semester	2017 2ND SEMESTER
	Class time and classroom		Mon 09:00~10:30 (Pal325)Tue 8.5(Pal328) Tue 9.5(Pal328)Wed 09:00~10:30 (Pal325)		English Grade	A(100%English)
Reference to this course	Prerequisite courses		Computer Programming, Discrete Mathematics			
	Related basic courses					
	Recommended concurrent courses					
	Related advanced courses					
Instructor	Name (title/division)		Yenewondim Biadgie.S(Assistant Professor, Software and Computer Engineering)			
	Office Room Number	팔달관 1011	Office phone Number	3857	e-mail	
	Office hours			Homepage address		
Teaching Assistant	Name (title/division)					
	Office Room Number		Office phone Number		e-mail	

1. Introduction

The course introduces basic data structures, like linked lists, stacks, queues, trees and sets. Each data structure has two representations. The first is logical representation and the second is physical representation in computer memory. Programing assignments using C language will be given to implement abstract data structures using array implementation or pointer implementation. Hence, first a review of basic concepts in c language such as Arrays, pointers, structures, static memory allocation and dynamic memory allocation will be given. Programming assignments enable students to compare the logical representation and the physical representation of each abstract data structure.

2. Course Objectives

To equip students with ability to

- ?Map non-linear data structures into physical memory which is linear in nature.
- ?Identify linear and nonlinear data structures
- ?Manipulate data structures with basic operations
- ?Compare different implementations of the same data structure.
- ?Choose a data structure which is efficient for a given application or problem.

3. Class types and activities

This class is based on multimedia lecture and active participation of students in problem solving. During lecture session, students are encouraged to be involved in problem-solving by asking questions before getting the correct answer from the instructor. Students will be involved actively in problem solving by implementing abstract Data structures. Students are supposed to spend considerable amount of time by doing programming assignments to understand this course.

4. Teaching Method

- | | |
|---|---|
| <input checked="" type="checkbox"/> lecture | <input checked="" type="checkbox"/> discussion and debate |
| <input checked="" type="checkbox"/> team project(presentation and case studies) | <input type="checkbox"/> experiments(role-playing,etc) |
| <input type="checkbox"/> designing and production | <input type="checkbox"/> on-site learning(on-site training) |
| <input type="checkbox"/> others | |

5. Support Systems in Use

- | | | |
|--|---|---|
| <input checked="" type="checkbox"/> e-class / AjouBb | <input type="checkbox"/> automatic recording system | <input type="checkbox"/> web-based assignment |
| <input type="checkbox"/> cyber lecture | <input type="checkbox"/> online content | |
| <input type="checkbox"/> class behavior analyzing system | <input type="checkbox"/> others | |

6. Teaching Tools

- | | | |
|--|---|---|
| <input type="checkbox"/> PBL(Problem Based Learning) | <input type="checkbox"/> CBL(Case Based Learning) | <input type="checkbox"/> TBL(Team Based Learning) |
| <input type="checkbox"/> UR(Undergraduate Research) | <input type="checkbox"/> FL(Flipped Learning) | <input type="checkbox"/> DSAL(Data Science Active Learning) |
| <input checked="" type="checkbox"/> others () | | |

7. Knowledge and ability required for taking this course

Computer Programming, Discrete Mathematics

8. Method of Evaluation

Evaluation Item	The Number of Times	Evaluation Proportion	Remarks
Attendance		5%	
midterm exam	1	30%	
final exam	1	35%	
quiz			
presentation			
discussion			
homework		30%	
etc			
study hours			

9. Textbook and supplementary material

Main/Sub	Title (Web-site)	Writer	Publisher	Publication year
Main	Fundamentals of Data Structure in C, 2nd ed.	Ellise Horowitz et al.	Silicon Press	2008

10. Class system and Class shedule

In the beginning of the course, concepts of algorithms, mathematical induction, and asymptotic analysis of an algorithm are taught. Algorithm design techniques follow including divide-and-conquer, dynamic programming, greedy method, and iterative improvements. Then students will learn problems that do not have efficient algorithms (NP-hard problems), and how to cope with such problems.

< Class Schedule >

* language : K-korean, E-English

Weeks	Topics	language	Instructor	Teaching Method	Evaluation Method	Matter to be prepared
1	Introduction to Algorithm Specification and Performance Analysis of an Algorithm	E	Yenewondim Biadgie.S			
2	Dynamic Arrays, Pointers and structures in C language	E	Yenewondim Biadgie.S			
3	Stack ADT	E	Yenewondim Biadgie.S			

< Class Schedule >

* language : K-korean, E-English

Weeks	Topics	language	Instructor	Teaching Method	Evaluation Method	Matter to be prepared
4	Queue ADT	E	Yenewondim Biadgie.S			
5	Linked ADT	E	Yenewondim Biadgie.S			
6	Tree(basic concepts and binary tree)	E	Yenewondim Biadgie.S			
7	Tree(binary search tree, heap)	E	Yenewondim Biadgie.S			
8	Midterm Exam	E	Yenewondim Biadgie.S			
9	Graphs (basic concepts, representation)	E	Yenewondim Biadgie.S			
10	Graphs(Shortest paths, Spanning Trees)	E	Yenewondim Biadgie.S			
11	Graphs(Activity-on-Arrow and Activity-on-Arrow networks)	E	Yenewondim Biadgie.S			
12	Hashing(Basic Concepts)	E	Yenewondim Biadgie.S			
13	Hashing(Static Hashing)	E	Yenewondim Biadgie.S			
14	Hashing(Dynamic Hashing)	E	Yenewondim Biadgie.S			
15	Summary and Review	E	Yenewondim Biadgie.S			
16	Final Exam	E	Yenewondim Biadgie.S			

11. Other items of notification