Course Syllabus-For Undergraduate Courses

Department of Industrial Engineering and Engineering Management

Course No.	IEEM		Required/Elective course		Required	
Course Time	M2M3M4 T2T3T4	Room	827		Size limit	20
Credits	3				Hours offered per week	6
Math	Basic Science	Engineering				
		Theory	Design			
1	0	1	1			
Course Title	資料結構 (Data Structure)					
Lecturer	Hareesh 何玄 (hareesh.pillai@ie.nthu.edu.tw)					
TA	葉力嘉 (home.yeh@gmail.com)					
Prerequisite	Basic programming constructs in any programming language.					

Core capability to be cultivated by this course	■	Integration-將工業工程與工程管理各項技術整合應用，有效處理工工領域相關議題 Abilities of integrating various technologies of Industrial Engineering and Engineering Management	20%
	■	Information -資訊科技的善加應用，以利解決工工領域之問題 Utilization of information technology to problem solving and applications in the field of Industrial Engineering	30%
	■	Interaction-協調溝通與團隊精神的發揮 Coordination and communication abilities with teamwork	20%
	■	Innovation/Ideas-激發創造力、培養創新思維 Creative and innovative capabilities	10%
	■	Internationalization-國際化互動與表達能力 Global interaction and public presentation skills	20%
Course Description	This course is designed for **undergraduate** students. Programs are comprised of two things: data and algorithms. The algorithms describe the way the data is to be transformed. The reason for learning about data structures is because adding structure to data can make the algorithms much simpler, easier to maintain, and often faster. The course is designed to practice concepts like arrays, linked list, queues, trees, graphs, and block chain. **Data structure through C** course is to understand the basic principles of data structures. C is often called a "Middle Level" programming language since it provides all the basic building blocks that are needed to produce results. The idea behind using C is to learn the nitty-gritty behind data structures from a core programming level along with an overview to object orientated class concepts using **Python** programming. Further, an overview of **Cython** that		

	combines the power of Python and C will be discussed for performance. This course emphasizes and illustrates the role of data structures that are employed in countless applications. There is an emphasis placed on real-world applications to engineering problems.
Textbook	Kanetkar, Y. P. (2016). *Let us C*. BPB publications. Horwitz, E., Sahni, S., & Anderson, S. (2008). Fundamentals of data structures in c. Kanetkar, Y. P. (2003). *Data Structures Through C*. BPB.
References	http://cython.org/
Teaching Method	The primary teaching objective will be towards enabling students to think logically and in context of real-world applications for data structures. The course is planned as a practice centered applications course.
Teaching software	PowerPoint, Turbo C/ GCC, and Python 3.X
Syllabus	Week 1. C Programming Refresher: Overview of C programming, running C programs, the structure of C programs, C's standard libraries, data types, language constructs, arrays, pointers, structures, unions. Week 2. Introduction to Data structures: Definition, classification of data structures (primitive and non-primitive), operations on data structures. Week 3. Dynamic memory allocation and pointers: Definition accessing the address of a variable, declaring and initializing pointers, accessing a variable through its pointer, meaning of static and dynamic memory allocation, memory allocation functions (malloc, calloc, free, and realloc). Week 4. Recursion: Definition, recursion in C, writing recursive programs – binomial coefficient, Fibonacci, GCD. Week 5 and 6. Searching, Sorting, Structures: Search techniques - Sequential search, binary search (iterative and recursive methods), comparison between sequential and binary search. Sort - General background, types (bubble sort, selection sort, merge sort, insertion sort, quick sort). Linked list – Definition, components of linked list, representation of linked list, advantages and disadvantages of linked list, types of linked list (singly linked list, doubly linked list, circular linked list and circular doubly linked list), operations on singly linked list (creation, insertion, deletion, search and display).

	Stack - Definition, array representation of stack, linked list representation of stack, operations on stack (infix, prefix and postfix notations), applications of stacks. Queue - Definition, array representation of queue, linked list representation of queue, types of queue: simple queue, circular queue, double ended queue (deque) priority queue, operations on all types of queues. Class- Definition, terminologies (structure, behavior, object), concepts (encapsulation, abstraction, polymorphism, inheritance) using Python. Block chain- Definition, terminologies (block, chain, hash), modeling and concepts using C, connecting through Cython. Tree – Definition (tree, binary tree, complete binary tree, binary search tree, heap tree), terminologies (root, node, degree of a node and tree, terminal nodes, non-terminal nodes, siblings, level, edge, path, depth, parent node, ancestors of a node), binary tree (array representation of tree, creation of binary tree), traversal of binary tree (preorder, inorder, and postorder). Graph – Definition, terminology (vertex, edge, adjacency, path), operations (add vertex, add edge, display vertex). Week 7. Real World Problem Solution Presentation: Week 8. Concluding Discussions:
Evaluation	The course includes weekly programming practice sessions mainly to apply data structure principles discussed in class, review material presented in class, and engage students in weekly 15-minute mini-quizzes. Along with these small quizzes, there is a final project. Grades will be based on individual and project performance. Individual grades come from class participation, contribution to the online discussion, and from individual quizzes. Students will also work on a work on a project related to mapping data structures for a real-world problem. Presenting in English is recommended alternatively Chinese oral presentation is allowed, however the PowerPoint slides and final report word file has to be in English. Individual performance: 50% Class and online participation – 20% Project – 30%
Course website	Lecture notes