ARSD College, University of Delhi

Model Course Handout/Lesson Plan

Course Name : B.Sc. (Hons.) Computer Science						
Semester	**Course Code**	**Course Title**	**Lecture (L)**	**Tutorial (T)**	**Practical (P)**	**Credit (C)**
III	32341301	Core-V (DSC) (BHCS05)- **Data Structures**	4	0	4	6
Teacher/Instructor(s)		Dr. Shalini Gupta				
Session		2022-23				

Course Objective:

This course aims at developing the ability to use basic data structures like array, stacks, queues, Linked lists, trees and hash tables to solve problems. C++ is chosen as the language to understand implementation of these data structures.

Course Learning Outcomes:

At the end of the course, students will be able to:

1. Implement and empirically analyze linear and non-linear data structures like Arrays, Stacks, Queues, Lists, Trees, Heaps and Hash tables as abstract data structures. (RBT L2/3)
2. Write a program, choosing a data structure, best suited for the application at hand. (RBT L3/4)
3. Re-write a given program that uses one data structure, using a more appropriate/ efficient Data Structures (RBT L4)
4. Write programs using recursion for simple problems. Explain the advantages and disadvantages of recursion.(RBT L2/L3)
5. Identify Ethical Dilemmas.

Lesson Plan:

Unit No.	Learning Objective	Lecture No.	Topics to be covered
1.	Arrays	1-2	single and multi-dimensional arrays
		3-4	analysis of insert, delete and search operations in arrays (both linear search and binary search)
		5-6	implementing sparse matrices
		7-8	applications of arrays to sorting: selection sort, insertion sort, bubble sort
		9-10	comparison of sorting techniques via

			empirical studies
		11-12	Introduction to Vectors.
2.	Linked Lists	13-14	Singly- linked Lists
		15-16	doubly-linked and circular lists
		17-18	analysis of insert, delete
		19-20	Analysis of search operations in all the three types
		21-22	implementing sparse matrices
		23-24	Introduction to Sequences
3.	Queues	25-26	Array and linked representation of queue
		27-28	de-queue, comparison of the operations on queues in the two representations
		29-30	Applications of queues.
4.	Stacks	31-32	Array and linked representation of stacks
		33-34	comparison of the operations on stacks in the two representations, implementing multiple stacks in an array
		35-36	applications of stacks: prefix, infix and postfix expressions, utility and conversion of these expressions from one to another;
		37-38	applications of stacks to recursion: developing recursive solutions to simple problems, advantages and limitations of recursion.
5.	Trees and Heaps	39-40	Introduction to tree as a data structure; binary trees, binary search trees
		41-42	analysis of insert, delete, search operations
		43-44	recursive and iterative traversals on binary search trees
		45-46	Height-balanced trees (AVL),
		47-48	B trees, analysis of insert, delete, search operations on AVL and B trees.
		49-50	Introduction to heap as a data structure. analysis of insert
		51-52	Analysis of extract-min/max and delete-min/max operations, applications to priority queues.
6	Hash Tables	53-54	Introduction to hashing, hash tables and hashing functions -insertion,
		55-56	Resolving collision by open addressing, deletion, searching and their analysis, properties of a good hash function.

Evaluation Scheme:

No.	Component	Duration	Marks
1.	Internal Assessment		25
	• Quiz		
	• Class Test		
	• Attendance		
	• Assignment		

2.	End Semester Examination	3 hrs	75

Details of the Course		
Unit	**Contents**	**Contact Hours**
I	**Arrays:** single and multi-dimensional arrays, analysis of insert, delete and search operations in arrays (both linear search and binary search), implementing sparse matrices, applications of arrays to sorting: selection sort, insertion sort, bubble sort, comparison of sorting techniques via empirical studies. Introduction to Vectors.	12
II	**Linked Lists:** Singly- linked, doubly-linked and circular lists, analysis of insert, delete and search operations in all the three types, implementing sparse matrices. Introduction to Sequences.	12
III	**Queues:** Array and linked representation of queue, de-queue, comparison of the operations on queues in the two representations. Applications of queues.	6
IV	**Stacks:** Array and linked representation of stacks, comparison of the operations on stacks in the two representations, implementing multiple stacks in an array; applications of stacks: prefix, infix and postfix expressions, utility and conversion of these expressions from one to another. Applications of stacks to recursion: developing recursive solutions to simple problems, advantages and limitations of recursion	8
V	**Trees and Heaps**: Introduction to tree as a data structure; binary trees, binary search trees, analysis of insert, delete, search operations, recursive and iterative traversals on binary search trees. Height-balanced trees (AVL), B trees, analysis of insert, delete, search operations on AVL and B trees. Introduction to heap as a data structure. analysis of insert, extract-min/max and delete-min/max operations, applications to priority queues.	14
VI	**Hash Tables**: Introduction to hashing, hash tables and hashing functions - insertion, resolving collision by open addressing, deletion, searching and their analysis, properties of a good hash function.	4
	Total	**56**

Suggested Books:

Sl. No.	**Name of Authors/Books/Publishers**	**Year of Publication/ Reprint**
1.	Drozdek, A., (2012), Data Structures and algorithm in C++. 4th edition. Cengage Learning	2012
2.	Goodrich, M., Tamassia, R., & Mount, D., (2011). Data Structures and Algorithms Analysis in C+ +. 2nd edition. Wiley.	2011

Mode of Evaluation:	Internal Assessment / End Semester Exam