University of Central Punjab
Faculty of Information Technology

PROGRAM (S) TO BE EVALUATED	**BSSE**

A. Course Description

Course Code	SECP2043
Course Title	Data Structures and Algorithms Lab
Credit Hours	4 (3 + 1)
Prerequisites	Object Oriented Programming
Assessment Instruments with Weights (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	Theory • Class Participation 10% • Graded Labs 30% • Midterm Tests 20% • Final 40%
Course Instructor	
Course Coordinator	Nabeel Sabir Khan
Office Hours	
Plagiarism Policy	Plagiarism (partial or full) is UNACCEPTABLE in this course. Collaboration and group work is encouraged but each student is required to submit his/her own contribution(s). Thus, ▪ Partial cheating is also cheating and is unacceptable ▪ All assignments must be done individually ▪ You may not copy code directly from any other source ▪ If you viewed another code (from books or lecture notes, web), you must include a reference in your assignment ▪ You may not share code with any other students by transmitting completed functions to your peers ▪ You may discuss assignment together and help another student debug his or her code; however, you cannot dictate or give the exact solution ▪ Collaboration with other students must be limited to discussions ▪ The minimum penalty for the first plagiarism is ZERO on the assessment and 5% penalty on your final grade ▪ The penalty is applied regardless of what proportion the assignments are of your final grade and regardless whether assessment is counted or not On one more attempt, the penalty is 'F', ranging from the module to the course.
Course Tools	Microsoft Visual Studio
Course Description	This course familiarizes students with concepts of creating, storing, retrieving, ordering, and manipulation of data structures and the basics of analysis of algorithms. The students will learn formal specification of data structures in depth. These goals will be accomplished by • Introducing the students to basic data structures and their associated algorithms. • Introducing the theory of complexity and develop the skills to analyze time and space requirements for a data structure and its associated

<table>
<tr><td></td><td colspan="4">algorithms.
Implementing data structures in C++ by determining, which data structures are appropriate in various situations.</td></tr>
<tr><td>Course Objectives</td><td colspan="4">On completion of this module, students should be able to:
• Understand the properties of various data structures.
• Identify the strengths and weaknesses of different data structures.
• Design and employ appropriate data structures for solving computing problems
• Possess the knowledge of various existing algorithms.
Analyze and compare the efficiency of algorithms.</td></tr>
<tr><td>Textbook</td><td colspan="4">1. D.S. Malik.C++ Programming: From Problem Analysis to Program Design, Cengage Learning; 8th edition
2. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++, Second/Third Edition, Edition, Addison Wesley.</td></tr>
<tr><td>Reference Material</td><td colspan="4"><u>Books:</u>
1. V. Aho, J. E. Hopcroft, J. D. Ullman, Data Structures and Algorithms, Addison-Wesley.
2. Thomas H. Cormen et al, Introduction to Algorithms, Prentice-Hall
3. Adam Drozdek, Data Structures and Algorithms in C++, Broooks/Cole
<u>Web References:</u>
1. http://courses.cs.vt.edu/csonline/DataStructures/Lessons/
2. http://www.cs.rutgers.edu/%7Ekaplan/503/
3. http://www.cplusplus.com/doc/tutorial/
4. http://www.cs.colorado.edu/~main/supplements/lectures.html
5. http://www.engr.mun.ca/~theo/Courses/ds/CPP-REV.HTM
6. http://www.nist.gov/dads/
7. http://www.cs.sunysb.edu/~skiena/214/lectures/</td></tr>
<tr><td>Topics Covered in the Course, with Number of Lectures on Each Topic (assume 15-week instruction and one-hour lectures)</td><td colspan="4">Attached</td></tr>
<tr><td>Programming Assignments Done in the Course</td><td colspan="4">Yes (C++ Programs)</td></tr>
<tr><td rowspan="2">Class Time Spent on (in credit hours)</td><td>Theory</td><td>Problem Analysis</td><td>Solution Design</td><td>Social and Ethical Issues</td></tr>
<tr><td>1</td><td>0.5</td><td>1</td><td>0.5</td></tr>
<tr><td>Oral and Written Communications</td><td colspan="4">Yes.</td></tr>
</table>

CLO	CLO STATEMENT	Level	PLO
1	**Understand** the properties of various data structures, including strengths and weaknesses of different data structures.	C2	1
2	**Analyze and compare** the efficiency of different data structures and algorithms.	C4	1
3	**Mapping** of appropriate data structures to various computing problems.	C4	3
4	Students will be able to **implement** different data structures and algorithms**.**	C3	2

CLO#	Course Learning Outcome (CLO)	Taxonomy Level	Mapping to PLO
CLO 1	Ability to sensibly select appropriate data structures and algorithms for problems and to justify that choice.		
CLO 2	Ability to estimate the algorithmic complexity of simple, non-recursive programs		
CLO 3	Ability to program data structures and use them in implementations of abstract data types.		
CLO 4	Understanding of fundamental Data Structures including linked-lists, trees, binary search trees, AVL trees, stacks, queues, priority queues, and hash-tables and skiplists		

Laboratory Outline

c	Lab#	Topics Covered	objective	Evaluation Used	Relation to CLO
1	**Lab#1:**	Revision	The basic purpose of this laboratory is revision of some preliminary concepts of C++ that has been covered in the course of Introduction to Computing and Object Oriented Programming. Its objective is to: • Array revision • Pointers • Static pointers • Dynamic pointers • Deference pointers	Class participation (lab tasks)	
2	**Lab#2:**	Searching and sorting algorithms	• Simple sorting algorithms Selection sort • Simple searching algorithms Linear search, binary search	Class participation (lab tasks)	C1
3	**Lab#3:**	Array List ADT Implementation	• Implementation of Array List ADT	Graded lab1	C1
4	**Lab#4:**	Linked Lists	• Implementation of LinkedLists ADT	Class participation (lab tasks)	C2
5	**Lab#5:**	Doubly and Doubly Circular Linked Lists	• Implementation of Doubly and Doubly Circular Linked Lists • Implementation of Stack and Queues using linked lists	Graded lab2	C2,C3
6	**Lab#6:**	Recursion	• Tower of Hanoi, basic problems (factorial, Fibonacci, decimal to binary, number of • digits in an integer)	Class participation (lab tasks)	C2,C3

7	**Lab#7:**	Applications of Stack Applications of Queues	• Working on Applications of Stack - Infix, Postfix and Prefix notations • Working on Applications of Queues – Priority Queues (Bank, Motorway e-tag)	Graded lab3	C2,C3
8	**Lab#8:**	Revision	Revision lab		C1, C2,C3
MID TERM WEEK					
9	**Lab#9:**	Binary Trees	Implementation of Binary Search Trees – different traversals, height of the tree, searching	Class participation (lab tasks)	C3
10	**Lab#10:**	Binary Search Trees	Applications of BST	Graded lab4	C3,C4
11	**Lab#11:**	Self-Balancing Trees	Implementation of AVL (Insertion with rotations)	Class participation (lab tasks)	C3,C4
12	**Lab#12:**	Self-Balancing Trees	Deletion, Update and Search AVL	Graded lab5	C3,C4
13	**Lab#13:**	Hash Tables	Applications using unordered maps continued	Class participation (lab tasks)	C4
14	**Lab#14:**	Heaps	Implementation of Min and Max Heaps	Graded lab6	C3,C4
15	**Lab#15:**	graphs	Implementation of graphs	Class participation (lab tasks)	C1,C2,C3, C4
16	**Lab#16:**	Revision	Revision Lab	Revision lab	C4
Final Term Week					