

<i>Open Elective</i>			
<b>Advanced C Programming Lab</b>			
Course Code	UCS559L/659L	CIE Marks	50
TeachingHours/Week (L:T:P)	0-2-2	SEE Marks	50
Credits	2	Hours	24
<b>The objective of the course is to:</b>			
<ul style="list-style-type: none"> <li>• Imbibe thorough knowledge in advanced C programming concepts.</li> <li>• Have proficiency in applying advanced C programming concepts to solve any real world problem.</li> </ul>			
<b>Unit -1 (6 hours)</b>			
Multidimensional arrays. Self-referential structures and Unions. <b>Pointers:</b> Introduction, Pointers for inter function communication, Pointers to pointers,			
<b>Revised Bloom's Taxonomy Level</b>	L <sub>1</sub> – Remembering, L <sub>2</sub> – Understanding, L <sub>3</sub> – Applying, L <sub>4</sub> – Analyzing, L <sub>5</sub> – Evaluating, L <sub>6</sub> – Creating		
<b>UNIT- II (6 hours)</b>			
<b>Pointer Applications:</b> Arrays and pointers, pointer arithmetic and arrays, passing an array to a function, memory allocation functions, array of pointers, Examples. Data Structures, Data structure Operations, Stacks: Definition, Stack Operations, Array Representation of Stacks.			
<b>Revised Bloom's Taxonomy Level</b>	L <sub>1</sub> – Remembering, L <sub>2</sub> – Understanding, L <sub>3</sub> – Applying, L <sub>4</sub> – Analyzing, L <sub>5</sub> – Evaluating, L <sub>6</sub> – Creating		
<b>UNIT- III (6 hours)</b>			
Stacks using Dynamic Arrays, Stack Applications: Queues: Definition, Array Representation, Queue Operations. Programming Examples.			
<b>Revised Bloom's Taxonomy Level</b>	L <sub>1</sub> – Remembering, L <sub>2</sub> – Understanding, L <sub>3</sub> – Applying, L <sub>4</sub> – Analyzing, L <sub>5</sub> – Evaluating, L <sub>6</sub> – Creating		
<b>UNIT- IV (6 hours)</b>			
<b>Linked Lists:</b> Definition, Representation of linked lists in Memory, Linked list operations: Traversing, Searching, Insertion, and Deletion. Applications of Linked lists. Implementation of stack and queue using linked list.			
<b>Revised Bloom's Taxonomy Level</b>	L <sub>1</sub> – Remembering, L <sub>2</sub> – Understanding, L <sub>3</sub> – Applying, L <sub>4</sub> – Analyzing, L <sub>5</sub> – Evaluating, L <sub>6</sub> – Creating		

**Course outcomes:**

*By the end of the course, the student will be able to:*

1. Define advanced C programming concepts like pointers, data structures.
2. Apply the knowledge of advanced C programming concepts to implement given requirement specification or to solve real world problem.
3. Analyze different data structures and use suitable data structure to implement requirement specification.
4. Implement, interpret, debug and test any given advanced C program.
5. Develop software product using advanced C programming concepts to solve real world problem.

Sl No	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
<b>Textbooks</b>				
1	Data Structures: A Pseudo-code approach with C	Gilberg&Foro uzan	Cengage Learning	2 <sup>nd</sup> Edition, 2014
2	Data Structures through C	Yashwant Kanetkar	BPB Publications	2017
<b>Reference Books</b>				
1	Data Structures: A Pseudo-code approach with C	Gilberg&Foro uzan	Cengage Learning	2 <sup>nd</sup> Edition, 2014
2	Data Structures using C	Reema Thareja	Oxford press	3 <sup>rd</sup> Edition 2012
3	An Introduction to Data Structures with Applications	Jean-Paul Tremblay & Paul G.	McGraw-Hill	2 <sup>nd</sup> Edition, 2013
<b>Web links and Video Lectures:</b> <a href="https://nptel.ac.in/courses/106/106/106106130/">https://nptel.ac.in/courses/106/106/106106130/</a> <a href="https://www.classcentral.com/course/edx-c-programming-pointers-and-memory-management-11533">https://www.classcentral.com/course/edx-c-programming-pointers-and-memory-management-11533</a> <a href="https://academicearth.org/computer-science/">https://academicearth.org/computer-science/</a> <a href="http://nptel.vtu.ac.in/econtent/courses/BS/15PCD23/index.php">http://nptel.vtu.ac.in/econtent/courses/BS/15PCD23/index.php</a>				

**Part A**

- 1** Write C program to accept and display 1D array Also write functions.
- to insert an element based at the specified position
  - to delete element based on the position
  - to delete based on the value
- function should take care of invalid data and accordingly display appropriate error messages.
- 2** Write C program to accept and display 2d array of user specified size. Also write functions to perform the following on the 2d array
- Function row\_sum that takes row number as parameter and returns the sum of the row
  - Function col\_sum that takes column number as parameter and returns the sum of the column
  - Function secondary\_diagonal\_sum that returns the sum of secondary diagonal elements if possible else should return -1
  - Function primary\_diagonal\_sum that returns the sum of primary diagonal elements if possible else should return -1
- 3.** Write C program to swap two integers using function.
- 4.** Write C program to accept and display 1d array. Use external pointer to process the array. Use separate functions to
- Accept the array elements
  - Display the array elements in forward direction
  - Display the array elements in reverse direction
  - To compute the average of the elements in the array
- 5.** Write C program to store information(name, employee\_id, designation, date of birth, stay details) about set of employees in a company. Here designation is string that can take one of these values {md, manager, clerk, peon} date\_of\_birth is a structure for holding birth date with fields day, month, year stay\_detail is a structure that contains street number and sector number and house number details. Write separate functions to accept & display the employees.

**Part - B**

1. Write C program to implement stack of integers using array.
2. Write C program to implement linear queue of integers using array.
3. Write C program to create & display singly linked list of integers.
4. Write C program to implement stack using linked list.
5. Write C program to implement queue using linked list.

**Course Articulation Matrix: Mapping of Course Outcomes (CO) with Programme Outcomes (PO) and Programme Specific Outcomes (PSO)**

		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
<b>No</b>	<b>Programme Outcomes Course Outcomes</b>															
<b>The students will be able to:</b>																
1	Define advanced C programming concepts like pointers, data structures.			1												
2	Apply the knowledge of advanced C programming concepts to implement given requirement specification or to solve real world problem.		2	2										3		
3	Analyze different data structures and use suitable data structure to implement requirement specification.	1	3	3									2	3		1
4	Implement, interpret, debug and test any given advanced C program.		3	3									2	3		2
5	Develop software product using advanced C programming concepts to solve real world problem		3	3	3								3	3		3