Ref No:

SRI KRISHNA INSTITUTE OF TECHNOLOGY
BANGALORE

-

COURSE PLAN

Academic Year FEB 2020

Program:	B E – Computer Science & Engineering
Semester :	4
Course Code:	18CSL47
Course Title:	Design and Analysis of Algorithms Lab
Credit / L-T-P:	2 / 0-0-2
Total Contact Hours:	36
Course Plan Author:	Rajesh/Sushma.M/Shilpa

Academic Evaluation and Monitoring Cell

No. 29, Chimney hills, Hesaraghatta Road, Chikkabanavara

INSTRUCTIONS TO TEACHERS

- Classroom / Lab activity shall be started after taking attendance.
- Attendance shall only be signed in the classroom by students.
- Three hours attendance should be given to each Lab.
- Use only Blue or Black Pen to fill the attendance.
- Attendance shall be updated on-line & status discussed in DUGC.
- No attendance should be added to late comers.
- Modification of any attendance, over writings, etc is strictly prohibited.
- Updated register is to be brought to every academic review meeting as per the COE.

Table of Contents

Note : Remove “Table of Content” before including in CP Book
Each Laboratory Plan shall be printed and made into a book with cover page
Blooms Level in all sections match with A.2, only if you plan to teach / learn at higher levels

A. LABORATORY INFORMATION

1. Laboratory Overview

Degree:	B E	*Program:*	CS
Year / Semester :	2/ 4	*Academic Year:*	2019-20
Course Title:	Design And Analysis Of Algorithm Laboratory	*Course Code:*	18CSL47
Credit / L-T-P:	3/ 01+02	*SEE Duration:*	3Hrs
Total Contact Hours:	36 Hrs	*SEE Marks:*	60 Marks
CIA Marks:	40	*Assignment*	-
Course Plan Author:	Rajesh/Sushma/Shilpa	*Sign*	Dt :
Checked By:		*Sign*	Dt :

2. Laboratory Content

Unit	**Title of the Experiments**	**Lab Hours**	**Concept**	**Blooms Level**
1.		3		
a.	Create a Java class called *Student* with the following details as variables within it. (i) USN (ii) Name (iii) Branch (iv) Phone Write a Java program to create *nStudent objects and* print the USN, Name, Branch, and Phone of these objects with suitable headings.		Classes and Objects	L3 Apply
b.	Write a Java program to implement the Stack using arrays. Write Push(), Pop(), and Display() methods to demonstrate its working.		Classes and Objects	L3 Apply
2.		3		
a	Design a superclass called *Staff* with details as StaffId, Name, Phone, Salary. Extend this class by writing three subclasses namely *Teaching* (domain, publications), *Technical* (skills), and *Contract* (period). Write a Java program to read and display at least 3 *staff* objects of all three categories.		Classes and Objects	L3 Apply
b	Write a Java class called *Customer* to store their name and date_of_birth. The date_of_birth format should be dd/mm/yyyy. Write methods to read customer data as <name, dd/mm/yyyy> and display as <name, dd, mm, yyyy> using StringTokenizer class considering the delimiter character as "/".		Classes and Objects	L3 Apply
3.		3		
a.	Write a Java program to read two integers *aandb*. Compute *a/b* and print, when *b* is not zero. Raise an exception when *b* is equal to zero.		Classes and Objects	L3 Apply
b.	Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer for every 1 second; second thread computes the square of the number and prints; third thread will print the value of cube of the number.		Classes and Objects	L3 Apply
4	Sort a given set of *n* integer elements using Quick Sort method and compute its time complexity. Run the program for varied values of	3	Divide &Conque	L4 Analyze

	n> 5000 and record the time taken to sort. Plot a graph of the time taken versus *n*on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.		r	
5.	Sort a given set of *n* integer elements using Merge Sort method and compute its time complexity. Run the program for varied values of *n*> 5000, and record the time taken to sort. Plot a graph of the time taken versus *n*on graph sheet. The elements can be read from a file or can be generated using the random number generator. Demonstrate using Java how the divide-and-conquer method works along with its time complexity analysis: worst case, average case and best case.	3	Divide &Conquer	L4 Analyze
6	Implement in Java, the 0/1 Knapsack problem using	3		L3 Apply
A	Dynamic Programming method		Dynamic Programming	
B	Greedy method		Greedy method	
7	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm. Write the program in Java.	3	Greedy method	L3 Apply
8	Find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal'salgorithm. Use Union-Find algorithms in your program.	3	Greedy method	L3 Apply
9	Find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.	3	Greedy method	L3 Apply
10	Write Java programs to	3	Dynamic Programming	L3 Apply
A	Implement All-Pairs Shortest Paths problem using Floyd's algorithm.			
B	Implement Travelling Sales Person problem using Dynamic programming.			
11	Design and implement in Java to find a subset of a given set S = {Sl, S2,.....,Sn} of *n* positive integers whose SUM is equal to a given positive integer *d*. For example, if S ={1, 2, 5, 6, 8} and *d*= 9, there are two solutions {1,2,6}and {1,8}. Display a suitable message, if the given problem instance doesn't have a solution.	3	Backtracking	L3 Apply
12	Design and implement in Java to find all Hamiltonian Cycles in a connected undirected Graph G of *n* vertices using backtracking principle.	3	Backtracking	L3 Apply

3. Laboratory Material

Books & other material as recommended by university (A, B) and additional resources used by Laboratory teacher (C).

Expt.	Details	Expt. in book	Availability
A	**Text books (Title, Authors, Edition, Publisher, Year.)**	-	-
1, 2, 3, 4, 5,10	1. Introduction to the Design and Analysis of Algorithms, Anany Levitin:, 2rd Edition, 2009.Pearson.	Available	In Lib / In Dept
6,7,8,	2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and	Availabl	In Lib/ In

9,11,12	Rajasekaran, 2nd Edition, 2014,Universities Press	e	dept
B	**Reference books (Title, Authors, Edition, Publisher, Year.)**	**-**	**-**
		?	In Lib
		?	Not Available
C	**Concept Videos or Simulation for Understanding**	**-**	**-**
C1			
C2			
D	**Software Tools for Design**	**-**	**-**
E	**Recent Developments for Research**	**-**	**-**
		?	In lib
F	**Others (Web, Video, Simulation, Notes etc.)**	**-**	**-**
1 2 3 4 5	https://www.cs.duke.edu/courses/fall08/cps224/Book.pdf http://www.cse.iitd.ernet.in/~ssen/csl356/root.pdf http://www.imsc.res.in/~vraman/pub/intro_notes.pdf http://www.ics.uci.edu/~goodrich/teach/cs161/notes/ http://elearning.vtu.ac.in/06CS43.html		
?			

4. Laboratory Prerequisites:

Refer to GL01. If prerequisites are not taught earlier, GAP in curriculum needs to be addressed. Include in Remarks and implement in B.5.

Students must have learnt the following Courses / Topics with described Content . . .

Expt.	Lab. Code	Lab. Name	Topic / Description	Sem	Remarks	Blooms Level
1	17CSL27	Programming in C	Data Types,arrays.strings	2		Understand L2
2	17CSL38	Datastructure	Knowledge on Data Structures	3		Understand L2
-						
-						

5. Content for Placement, Profession, HE and GATE

The content is not included in this course, but required to meet industry & profession requirements and help students for Placement, GATE, Higher Education, Entrepreneurship, etc. Identifying Area / Content requires experts consultation in the area.

Topics included are like, a. Advanced Topics, b. Recent Developments, c. Certificate Courses, d. Course Projects, e. New Software Tools, f. GATE Topics, g. NPTEL Videos, h. Swayam videos etc.

Expt.	Topic / Description	Area	Remarks	Blooms Level
1				

3				
3				
5				
-				

B. Laboratory Instructions

1. General Instructions

SNo	Instructions	Remarks
1	Observation book and Lab record are compulsory.	
2	Students should report to the concerned lab as per the time table.	
3	After completion of the program, certification of the concerned staff in-charge in the observation book is necessary.	
4	Student should bring a notebook of 100 pages and should enter the readings /observations into the notebook while performing the experiment.	
5	The record of observations along with the detailed experimental procedure of the experiment in the Immediate last session should be submitted and certified staff member in-charge.	
6	Should attempt all problems / assignments given in the list session wise.	
7	It is responsibility to create a separate directory to store all the programs, so that nobody else can read or copy.	
8	When the experiment is completed, student should save the experiment with relevant filenames and exit from the Turbo C IDE compiler.	
9	Any damage of the equipment of the computer system will be viewed seriously either by putting penalty or by dismissing the total group of students from the lab for the semester/year	
10	Completed lab assignments should be submitted in the form of a Lab Record in which you have to write the algorithm, Flowchart, program code along with comments and output for various inputs given	

2. Laboratory Specific Instructions

SNo	Specific Instructions	Remarks
1	Start windows Operating system	
2	Open the eclipse Juno IDE in Windows	
3	To create a project: 1. On the main menu bar, click *File* -> *New Project*. The New Project wizard opens. 2. Select a category from the left column and then select the type of project to create from the right column. To assist in locating a particular wizard, the text field can be used to show only the wizards that match the entered text. Click Next. 3. In the Project name field, type a name for your new project. 4. (Optional) The project that you create will map to a directory structure in the file system. The default file system location is displayed in the Location field. If you want to create the project and its contained resources in a different location, clear the Use default location checkbox and specify the new location. 5. Click Finish. The new project is listed in one of the navigation views.	
4	To create a file:	

	1. In one of the navigation views, right-click the project or folder where you want to create the new file. 2. From the pop-up menu, select New -> File. 3. Specify the name of the file, including the file extension (for example, newfile.java). 4. Click Finish.	
5	Type the program	
6	Debug the program	
7	Execute the Program	

C. OBE PARAMETERS

1. Laboratory Outcomes

Expt.	Lab Code #	COs / Experiment Outcome	Teach. Hours	Concept	Instr Method	Assessment Method	Blooms' Level
-	-	**At the end of the experiment, the student should be able to . . .**	-	-	-	-	-
1	18CSL47.1	Develop java programs to demonstrate Stack Operation,inheritance, String Tokenized, Exception handling and Multi threading.	07	Classes and Objects/Stack Operation/Inheritance/ Multi-threading	Demonstrate	Viva & presentation	L5 Evaluate
2	18CSL47.2	Analyze the time efficiencies of sorting method using Divide and conquer	06	Quick Sort/ Merge Sort	Demonstrate	Viva & presentation	L5 Evaluate
3	18CSL47.3	Judge knapsack problems using greedy and dynamic programming.	3	Knapsack	Demonstrate	Viva & presentation	L5 Evaluate
4	18CSL47.4	Judge Shortest path and minimum spanning tree by implementing Dijkstra, Prims and kruskal's using greedy Techniques.	7	Shortest path/minimum Spanning tree	Demonstrate	Viva & presentation	L5 Evaluate
5	18CSL47.5	Implement Shortest path and Shortest distance by implementing Floyd's algorithm and TSP methods using dynamic programming technique. Appraise generating subset and Hamiltonian cycle using backtracking	6	Dynamic Programming, Generating subset / Hamiltonian cycle	Demonstrate	Viva & presentation	L5 Evaluate
-		**Total**	**36**	**-**	**-**	**-**	**-**

Note: Identify a max of 2 Concepts per unit. Write 1 CO per concept.

2. Laboratory Applications

Expt.	Application Area	CO	Level
1	Multiprocessor computers	CO1	L5
2	Text editors,web browsers	CO1	L5
3	Image processing	CO2	L5
4	Optimization problem	CO2	L5
5	Huffman trees	CO3	L5

6	Mind games, puzzles.	CO3	L5
7	Evaluate traveling sales man problem by using dynamic programming	CO4	L3
8	Apply Branch and Bound for solving combinatorial optimization problems	CO4	L2
9	Able to differentiate NP – Hard and NP – Complete Problems	CO5	L2

Note: Write 1 or 2 applications per CO.

3. Mapping And Justification

CO – PO Mapping with mapping Level along with justification for each CO-PO pair.
To attain competency required (as defined in POs) in a specified area and the knowledge & ability required to accomplish it.

Expt.	Mapping		Mapping Level	Justification for each CO-PO pair	Level
-	**CO**	**PO**	-	**'Area': 'Competency' and 'Knowledge' for specified 'Accomplishment'**	-
1,2,3	CO1	PO1	3	The knowledge of structure and abstract data type can be applied to solve complex problems.	L6
4,5	CO2	PO2	3	These fundamental concepts of CS can be applied to solve complex problems	L4
		PO3	3	Efficient algorithms can be designed based on their time complexity.	L6
6	CO3	PO2	3	These fundamental concepts of CS can be applied to solve complex problems	-
		PO3	3	Efficient algorithms can be designed based on their time complexity.	
7,8,9	CO4	PO1	3	The knowledge of structure and abstract data type can be applied to solve complex problems.	L6
		PO3	3	Efficient algorithms can be designed based on their time complexity.	-
		PO4	3	Analysis of algorithms helps to select suitable algorithms and reach valid conclusions.	-
10,11,12	CO5	PO1	3	The knowledge of structure and abstract data type can be applied to solve complex problems.	L6
		PO2	3	These fundamental concepts of CS can be applied to solve complex problems	-
		PO3	3	Efficient algorithms can be designed based on their time complexity.	-
		PO4	3	Analysis of algorithms helps to select suitable algorithms and reach valid conclusions.	-

4. Articulation Matrix

CO – PO Mapping with mapping level for each CO-PO pair, with course average attainment.

-	-	Experiment Outcomes	Program Outcomes															-
Expt.	CO.#	**At the end of the experiment student should be able to . . .**	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	Lev el
1,2,3	18CSL47.1	Develop java programs to demonstrate Stack Operation,inheritance, String Tokenized, Exception handling and Multi threading.	3	3	3	-	3	-	-	-	-	-	-	3	-	-	-	L5
4,5	18CSL47.2	Analyze the time efficiencies of sorting method using Divide and conquer	3	3	3	-	3	-	-	-	-	-	-	3	-	-	-	L5
6	18CSL47.3	Judge knapsack problems using greedy and dynamic programming.	3	3	3	-	3	-	-	-	-	-	-	3	-	-	-	L5
7,8,9	18CSL47.4	Judge Shortest path and minimum spanning tree by implementing Dijkstra, Prims and kruskal's using greedy	3	3	3	-	3	-	-	-	-	-	-	3	-	-	-	L5

		Techniques.																
10,11, 12	18CSL47.5	Implement Shortest path and Shortest distance by implementing Floyd's algorithm and TSP methods using dynamic programming technique. Appraise generating subset and Hamiltonian cycle using backtracking. Appraise generating subset and Hamiltonian cycle using backtracking. Appraise generating subset and Hamiltonian cycle using backtracking.	3	3	3	-	3	-	-	-	-	-	-	3	-	-	-	L5
-	**18CSL47**	**Average attainment (1, 2, or 3)**	3	3	3	-	3	-	-	-	-	-	-	3	-	-	-	**-**
-	*PO, PSO*	*1.Engineering Knowledge; 2.Problem Analysis; 3.Design / Development of Solutions; 4.Conduct Investigations of Complex Problems; 5.Modern Tool Usage; 6.The Engineer and Society; 7.Environment and Sustainability; 8.Ethics; 9.Individual and Teamwork; 10.Communication; 11.Project Management and Finance; 12.Life-long Learning; S1.Software Engineering; S2.Data Base Management; S3.Web Design*																

5. Curricular Gap and Experiments

Topics & contents not covered (from A.4), but essential for the course to address POs and PSOs.

Expt	Gap Topic	Actions Planned	Schedule Planned	Resources Person	PO Mapping
1					
2					
3					
4					
5					

Note: Write Gap topics from A.4 and add others also.

6. Experiments Beyond Syllabus

Topics & contents required (from A.5) not addressed, but help students for Placement, GATE, Higher Education, Entrepreneurship, etc.

Expt	Gap Topic	Actions Planned	Schedule Planned	Resources Person	PO Mapping
1					
2					
3					
4					

D. COURSE ASSESSMENT

1. Laboratory Coverage

Assessment of learning outcomes for Internal and end semester evaluation. Distinct assignment for each student. 1 Assignment per chapter per student. 1 seminar per test per student.

Unit	Title	Teaching Hours	No. of question in Exam							CO	Levels
			CIA-1	CIA-2	CIA-3	Asg-1	Asg-2	Asg-3	SEE		
1	Student Class and Object Creation using Java	1.5	1	-	1	-	-	-	1	CO1	L5
2	Stack	1.5	1		1					CO1	
3	Staff Database	1.5	1	-	1	-	-	-	1	CO1	L5

4	Customer data	1.5	1	-	1	-	-	-	1	CO1	L5
5	Compute *a/b*	1.5	1	-	1	-	-	-	1	CO1	L5
6	Multi thread application	1.5	1	-	1	-	-	-	1	CO1	L5
7	Quick sort	03	1	-	1	-	-	-	1	CO2	L5
8	Merge sort	03	1	-	1	-	-	-	1	CO2	L5
9	0/1 Knapsack problem using Dynamic Programming	1.5	1	-	1					CO3	
10	0/1 Knapsack problem using Greedy Method	1.5	1	-	1	-	-	-	1	CO4	L5
11	Shortest Path using Dijkstra's algorithm	3	-	1	1	-	-	-	1	CO4	L5
12	Minimum Cost Spanning Tree using Kruskal's algorithm	3	-	1	1	-	-	-	1	CO4	L5
13	Minimum Cost Spanning Tree using prims algorithm	3	-	1	1	-	-	-	1	CO4	L5
14	All-Pairs Shortest Paths problem	1.5	-	1	1	-	-	-	1	CO3	L5
15	Traveling Sales Person problem	1.5	-	1	1	-	-	-	1	CO3	L5
16	Sum of subset problem	3	-	1	1					CO5	L5
17	Hamiltonian Cycles	3	-	1	1				1	CO5	L5
-	**Total**	**36**	**7**	**8**	**5**	**5**	**5**	**5**	**20**	**-**	**-**

2. Continuous Internal Assessment (CIA)

Assessment of learning outcomes for Internal exams. Blooms Level in last column shall match with A.2.

Evaluation	Weightage in Marks	CO	Levels
CIA Exam – 1	40	CO1,CO2,CO3,CO4	L3,L4
CIA Exam – 2	40	CO3,CO4,CO5	L3
CIA Exam – 3	40	CO1,CO2,CO3,CO4,CO5	L3,L4
Assignment - 1	-	-	-
Assignment - 2	-	-	-
Assignment - 3	-	-	-
	-	-	-
Seminar - 1	-	-	-
Seminar - 2	-	-	-
Seminar - 3	-	-	-
	-	-	-
Other Activities – define – Slip test	-	-	-
Final CIA Marks	**40**	**-**	**-**

-

SNo	Description	Marks
1	Observation and Weekly Laboratory Activities	05 Marks
2	Record Writing	15 Marks for each Expt
3	Internal Exam Assessment	20Marks
4	Internal Assessment	40 Marks
5	SEE	60Marks
-	**Total**	**100 Marks**

E. EXPERIMENTS

Experiment 1a: Student Class and Object Creation using Java

-	**Experiment No.:**	1a	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	A Java program to create *nStudent* objects and print the USN, Name, Branch, and *Phoneof* these objects with suitable headings.						
2	Course Outcomes	Develop java programs to demonstrate Inheritance, Exception handling and Multithreading.						
3	Aim	Create a Java class called Student with the following details as variables within it. • USN • Name • Branch • Phone create *nStudent* objects and print the USN, Name, Branch and Phone of these objects with suitable headings.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Object oriented Concepts						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	1. Create a stuent class with arguments to the constructor is USN, Name, Branch, Phone 2. Read the number of student objects to be created. 3. Read each student object details (USN, Name, Branch, Phone) 4. Display the USN, Name, Branch, and Phone number of each Student						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	-						
8	Observation Table, Look-up Table, Output	enter the no. of students 2 enter student details enter student name krishna enter student usn 1KT17IS12						

		enter student branch ISE enter student ph.no 9004565467 enter student name Hema enter student usn 1KT17IS18 enter student branch CSE enter student ph.no 9884543678 USN name branch phone 1KT17IS12 krishna ISE 9004565467 1KT17IS18 Hema CSE 9884543678
9	Sample Calculations	-
10	Graphs, Outputs	-
11	Results & Analysis	-
12	Application Areas	Computer Science
13	Remarks	-
14	Faculty Signature with Date	-

Experiment 1b : Stack

-	**Experiment No.:**	1b	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	A Java program to implement the Stack using arrays. Write Push(), Pop(), and Display()methods to demonstrate its working.						
2	Course Outcomes	Develop java programs to demonstrate Inheritance, Exception handling and Multithreading.						
3	Aim	Implementation of stack operations						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Push Operations Pop Operations Display Operations						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	Step 1: Start. Step 2: Initialize stack size MAX and top of stack -1. Step 3: Push integer element on to stack and display the contents of the stack. if stack is full give a message as 'Stack is Overflow'. Step 3: Pop element from stack along with display the stack contents. if stack is empty give a message as 'Stack is Underflow'. Step 4: Check whether the stack contents are Palindrome or not. Step 5: Stop.						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output	press 1 to push element press 2 to pop element press 3 to display elements press 4 to exit Enter your choice: 1 Enter element: 10						

```
The 10is pushed into the stack
Enter your choice:
1
Enter element:
20
The 20 is pushed into the stack
Enter your choice:
1
Enter element:
30
The 30is pushed into the stack
Enter your choice:
1
Enter element:
40
Error !Stack Overflow
Enter your choice:
3
Elements in stack
10
20
30
Enter your choice:
2
The 30 is poped out of the stack
Enter your choice:
2
The 20 is poped out of the stack
Enter your choice:
2
The 10 is poped out of the stack
Enter your choice:
2
error stack underflow
Enter your choice:
3
Stack Empty
Enter your choice:
4
Program stopped
```

9	Sample Calculations	Pushing the elements Poping the elements Checking the stack content form Palindrome Check overflow and underflow conditions
10	Graphs, Outputs	-
11	Results & Analysis	-
12	Application Areas	Code and debug the operations of stack
13	Remarks	
14	Faculty Signature with Date	

Experiment 2a : Staff Database

-	**Experiment No.:**	2a	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	**Write a Java program to read and display at least 3 *staff* objects of all three categories.**						
2	Course Outcomes	Develop java programs to demonstrate Inheritance, Exception handling and Multithreading.						

3	Aim	Understanding the concepts of inheritance and accessing the members of super class and sub class
4	Material / Equipment Required	Lab Manual
5	Theory, Formula, Principle, Concept	Object oriented Concepts
6	Procedure, Program, Activity, Algorithm, Pseudo Code	Step1:create a Super calss with the name staff and define required parameter Step2. Create subcalss named Teaching and extend the super class staff facilities Step3. Create subclass named Technical and extend the super class staff facilities Step4. Create the subclass Contract and extend the super class facilities staff facilities
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	Enter your category:1. Teaching, 2. Technical. 3.Contract 1 Enter SID,Salary,Name,Phone,domain and Publications 001 50000 Arun 98888888888 Android Journal Enter SID,Salary,Name,Phone,domain and Publications 002 200000 Manoj 78888888888 Netwoking International Enter SID,Salary,Name,Phone,domain and Publications 003 300000 Vinay 877777777 Cloud Journal Staff ID:1 Salary:50000 Name:98888888888 Phone:Arun Domain:Android Publication:Journal ----------------------------- Staff ID:2 Salary:200000 Name:78888888888 Phone:Manoj Domain:Netwoking Publication:International ----------------------------- Staff ID:3 Salary:300000 Name:877777777 Phone:Vinay Domain:Cloud

		Publication:Journal ------------------------------
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 2b : Customer data

-	**Experiment No.:**	2b	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Write a Java class called *Customer* to store their name and date_of_birth.						
2	Course Outcomes	Develop java programs to demonstrate Inheritance, Exception handling and Multithreading.						
3	Aim	Understanding the concepts of StringTokenizer class and separating the strings on the basis of different delimiters.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	String Tokenizer						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	1. Create a Java class Customer and define the required features 2. Read the date of birth in the prescribed format 3. Create a method to read the date string 4. Use StringTokenizer java class to tokenize the date string and print						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output	Enter Customer name: vikram Enter Customer DOB in the format dd/mm/yyy 01/01/1990 Customer Details................ vikram,01,01,1990						
9	Sample Calculations							
10	Graphs, Outputs							
11	Results & Analysis							
12	Application Areas							
13	Remarks							
14	Faculty Signature with Date							

Experiment 3a : Compute *a/b.*

-	**Experiment No.:**	5	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	A Java program to read two integers *a* and *b*. Compute *a/b* and print, when *b* is not zero.*Raise an exception when b is equal to zero.*						
2	Course Outcomes	Develop java programs to demonstrate Inheritance, Exception handling and Multithreading.						

3	Aim	Compute a/b and print, when b is not zero. Raise an exception when b is equal to zero.
4	Material / Equipment Required	Lab Manual
5	Theory, Formula, Principle, Concept	Object Oriented Concepts
6	Procedure, Program, Activity, Algorithm, Pseudo Code	1. Read two intergers a and b 2. Compute division a/b 3. If b is not zero print the result without exception 4. If b = 0 print the exception by using Java maths exceptions
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	Sample 1: Please enter first number (numerator): 10 Please enter second number (denominator): 5 Division result of 10/5= 2.0 Sample2: Please enter first number (numerator): 10 Please enter second number(denominator): 0 Exception Condition Program is ending
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 3b : Multithread application using Java

-	**Experiment No.:**	3b	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Multithread application						
2	Course Outcomes	Develop java programs to demonstrate Inheritance, Exception handling and Multithreading.						
3	Aim	To understand the concepts of multithreading by creating three threads that perform different tasks when one thread is suspended for some time duration.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Object Oriented Concepts						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	1. Create a class named multithread. 2. Create three thread using thread library. 3. First thread is for generating random integer. 4. Second thread is for square of the number generated by first thread. 5. Thrid thread compute the cube of the number generated by first.						

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	first thread generated number is77 Second thread:Square of the number is5929 third thread:Cube of the number is456533 first thread generated number is76 Second thread:Square of the number is5776 third thread:Cube of the number is438976 first thread generated number is14 Second thread:Square of the number is196 third thread:Cube of the number is2744
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 04 : Quick sort

-	**Experiment No.:**	4	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Quick sort						
2	Course Outcomes	Analyze and compare the performance of algorithms using language features.						
3	Aim	To sort 'n' randomly generated elements using Quick sort and plotting the graph of the time taken to sort n elements versus n.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code	1. Declare time variables 2. Generate 'n ' elements randomly using random number generator 3. Record start time before sorting 4. Call Quick sort function to sort n elements 5. Record the end time after sorting 6. Calculate the time required to sort n elements using Quick sort. 7. Print the sorted ' n' elements and time taken to sort. 8. Repeat the above steps for different values of n as well as to demonstrate worst, best and average case complexity.						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output	.Enter the no. of elements or Array Size > 5000 5010 The array elements before sorting are: 2613 543 3551 3898 3914 2880 671 2303 336 1273 ……………………….. ***********Quick Sort Algorithm ***************** The array elements after sorting are: 336 543 671 1273 2303 2613 2880 3551 3898 3914……… The time taken to sort is:1ms						

9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 05 : Merge Sort

-	**Experiment No.:**	7	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Merge Sort						
2	Course Outcomes	Analyze and compare the performance of algorithms using language features.						
3	Aim	To sort 'n' randomly generated elements using Merge sort and plotting the graph of the time taken to sort n elements versus n.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Divide & Conquer						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	1. Declare time variables 2. Generate 'n ' elements randomly using random number generator 3. Record start time before sorting 4. Call Quick sort function to sort n elements 5. Record the end time after sorting 6. Calculate the time required to sort n elements using Quick sort. 7. Print the sorted ' n' elements and time taken to sort. 8. Repeat the above steps for different values of n as well as to demonstrate worst, best and average case complexity.						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output	**Enter the no. of elements or Array Size > 5000** 5010 The array elements before sorting are: 2613 543 3551 3898 3914 2880 671 2303 336 1273 ……………………….. ***********Quick Sort Algorithm ***************** The array elements after sorting are: 336 543 671 1273 2303 2613 2880 3551 3898 3914……… The time taken to sort is:2ms						
9	Sample Calculations							
10	Graphs, Outputs							
11	Results & Analysis							
12	Application Areas	Image Processing						
13	Remarks							
14	Faculty Signature with Date							

Experiment 6a : 0/1 Knapsack problem using Dynamic Programming

-	Experiment No.:	8	Marks		Date Planned		Date Conducted	
1	Title	0/1 Knapsack problem using Dynamic Programming						
2	Course Outcomes	Demonstrate Dynamic Programming using 0/1 Knapsack,Floyd's Algorithm and Travelling Sales Person problem,						
3	Aim	To choose the set of items that fits in the knapsack and maximizes the profit. Given a knapsack with maximum capacity *W*, and a set *S* consisting of *n* items.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Dynamic Programming						

6	Procedure, Program, Activity, Algorithm, Pseudo Code	//Input: (n items, W weight of sack) Input: n, wi,,, vi and W – all integers
		//Output: V(n,W)
		Steps:
		// Initialization of first column and first row elements
		• Repeat for i = 0 to n
		set V(i,0) = 0
		• Repeat for j = 0 to W
		Set V(0,j) = 0
		//complete remaining entries row by row
		• Repeat for i = 1 to n
		repeat for j = 1 to W
		if (wi <= j) V(i,j)) = max{ V(i-1,j), V(i-1,j-wi) + vi }
		if (wi > j) V(i,j) =V(i-1,j)
		• Print V(n,W)						

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output	Enter the number of elements 5 Enter the profits of the element10 15 20 25 30 Enter the weight of the elements 3 4 5 2 1 Enter the the capacity of knapsack : 7 the profit gained is:70 Items selected:2 4 5						
9	Sample Calculations							
10	Graphs, Outputs							
11	Results & Analysis							
12	Application Areas	Image Processing						
13	Remarks							
14	Faculty Signature with Date							

Experiment 6b : 0/1 Knapsack problem using Greedy method

-	Experiment No.:	9	Marks		Date Planned		Date Conducted	
1	Title	0/1 Knapsack problem using Greedy method						
2	Course Outcomes	Demonstrate Greedy method using 0/1 Knapsack,Dijkstra's Algorithm,Kruskal's Algorithm and prims algorithm						
3	Aim	To choose the set of items that fits in the knapsack and maximizes the profit.						

		Given a knapsack with maximum capacity *W*, and a set *S* consisting of *n* items.
4	Material / Equipment Required	Lab Manual
5	Theory, Formula, Principle, Concept	Greedy method
6	Procedure, Program, Activity, Algorithm, Pseudo Code	Assume knapsack holds weight W and items have value vi and weight wi • Rank items by value/weight ratio: vi / wi Thus: vi / wi ≥ vj / wj, for all i ≤ j • Consider items in order of decreasing ratio • Take as much of each item as possible based on knapsack's capacity
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	-
8	Observation Table, Look-up Table, Output	Enter no of items 4 Enter the weights of each items 5 10 15 20 Enter the profits of each items 12 13 14 15 Enter capacity of knapsack : 18 Quantity of item number: 1 added is 5 Quantity of item number: 2 added is 10 Quantity of item number: 3 added is 3 The total profit is 27.8
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 7 : Shortest Path using Dijkstra's algorithm

-	**Experiment No.:**	10	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Shortest Path using Dijkstra's algorithm						
2	Course Outcomes	Demonstrate Greedy method using 0/1 Knapsack,Dijkstra's Algorithm,Kruskal's Algorithm and prims algorithm						
3	Aim	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Greedy method						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	function Dijkstra(*Graph*, *source*): 1 create vertex set Q 2 for each vertex *v* in *Graph*: *// Initialization* 3 dist[*v*] ← INFINITY *// Unknown distance from source to v*						

		prev[*v*] ← UNDEFINED *//previous node in optimal path from source* add *v* to *Q* *// All nodes initially in Q (unvisited nodes)* dist[*source*] ← 0 *// Distance from source to source* while *Q* is not empty: *u* ← vertex in *Q* with min dist[u]*//Node with the least distance* *// will be selected first* remove *u* from *Q* for each neighbor *v* of *u*: *// where v is still in Q.* *alt* ← dist[*u*] + length(*u*, *v*) if *alt* < dist[*v*]: *// A shorter path to v has been found* dist[*v*] ← *alt* prev[*v*] ← *u* return dist[], prev[]
	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	enter the no. of vertices 6 enter the cost of edges enter 999 if the edges are not present or selfloop 0 15 10 999 45 999 999 0 15 999 20 999 20 999 0 20 999 999 999 10 999 0 35 999 999 999 999 30 0 999 999 999 999 4 999 0 enter the source vertex 6 source destination cost 6---------1 49 6---------2 14 6---------3 29 6---------4 4 6---------5 34 6---------6 0
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 8 :Minimum Cost Spanning Tree using Kruskal's algorithm

-	**Experiment No.:**	8	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Minimum Cost Spanning Tree using Kruskal's algorithm						
2	Course Outcomes	Demonstrate Greedy method using 0/1 Knapsack,Dijkstra's Algorithm,Kruskal's Algorithm and prims algorithm						
3	Aim	Find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula,	Greedy Method						

	Principle, Concept	

6	Procedure, Program, Activity, Algorithm, Pseudo Code	ALGORITHM:
		KRUSKAL(G):
		A = Ø
		foreach v ∈ G.V:
		MAKE-SET(v)
		foreach (u, v) in G.E ordered by weight(u, v), increasing:
		if FIND-SET(u) ≠ FIND-SET(v):
		A = A ∪ {(u, v)}
		UNION(u, v)
		return A

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	Enter the number of nodes: 4 Enter the adjacency matrix: 999 20 10 999 20 999 999 30 10 999 999 40 999 30 40 999 Edge1: 1→3 cost:10 Edge2: 1→2 cost:20 Edge3: 2→4 cost:30 Minimun cost=60
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 9 :Minimum Cost Spanning Tree using Prims Algorithm

-	**Experiment No.:**	12	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Minimum Cost Spanning Tree using Prims Algorithm						
2	Course Outcomes	Demonstrate Greedy method using 0/1 Knapsack,Dijkstra's Algorithm,Kruskal's Algorithm and prims algorithm						
3	Aim	Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Greedy method						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	ALGORITHM: MST-PRIM(G, w r) for each u ∈ G.V u.key = ∞ u.π = NIL r.key = 0						

		Q = Q.V
		while Q = φ
		u = EXTRACT-MIN(Q) //minimum priority queue
		for each v ∈ G.Adj(u)
		v ∈ Q and w(u, v) < v.key
		v.π = u
		v.key = w(u, v)

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	Enter the adjacency matrix: 999 20 10 999 20 999 999 30 10 999 999 40 999 30 40 999 Edge1: 1→3 cost:10 Edge2: 1→2 cost:20 Edge3: 2→4 cost:30 Minimun cost=60
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 10 a : All-Pairs Shortest Paths problem

-	Experiment No.:	13	Marks		Date Planned		Date Conducted	

1	Title	All-Pairs Shortest Paths problem
2	Course Outcomes	Demonstrate Dynamic Programming using 0/1 Knapsack,Floyd's Algorithm and Travelling Sales Person problem,
3	Aim	Implement All-Pairs Shortest Paths problem using Floyd's algorithm
4	Material / Equipment Required	Lab Manual
5	Theory, Formula, Principle, Concept	Greedy Method

6	Procedure, Program, Activity, Algorithm, Pseudo Code	**ALGORITHM:**
		let dist be a |V| × |V| array of minimum distances initialized to ∞ (infinity)
		for each edge (*u*,*v*)
		dist[*u*][*v*] ← w(*u*,*v*) *// the weight of the edge* (u,v)
		for each vertex *v*
		dist[*v*][*v*] ← 0
		for *k* from 1 to |V|
		for *i* from 1 to |V|
		for *j* from 1 to |V|
		if dist[*i*][*j*] > dist[*i*][*k*] + dist[*k*][*j*]
		dist[*i*][*j*] ← dist[*i*][*k*] + dist[*k*][*j*]
		end i

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table,	Enter the no. of vertices

	Look-up Table, Output	4 Enter the weight matrix 0 999 3 999 2 0 999 999 999 7 0 1 6 999 999 0 all pair shortest path: 0 10 3 4 2 0 5 6 7 7 0 1 6 16 9 0
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 10 b :Travelling Sales Person problem

-	Experiment No.:	14	Marks		Date Planned		Date Conducted	
1	Title	Travelling Sales Person problem						
2	Course Outcomes	Demonstrate Dynamic Programming using 0/1 Knapsack,Floyd's Algorithm and Travelling Sales Person problem,						
3	Aim	To find the shortest possible route that visits every city exactly once and returns to the starting point.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Dynamic Programming						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	Algorithm: Traveling-Salesman-Problem C ({1}, 1) = 0 for s = 2 to n do for all subsets S Є {1, 2, 3, … , n} of size s and containing 1 C (S, 1) = ∞ for all j Є S and j ≠ 1 C (S, j) = min {C (S – {j}, i) + d(i, j) for i Є S and i ≠ j} Return minj C ({1, 2, 3, …, n}, j) + d(j, i)						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output	Enter No. of Cities: 4 Enter the Cost Matrix 0 10 15 20 5 0 9 10 6 13 0 12 8 8 9 0 The Cost Matrix is 0 10 15 20 5 0 9 10 6 13 0 12 8 8 9 0						

		The Optimal Tour is = 1->2->4->3->1 Minimum Cost = 35
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 11: Sum of Subset Problem

-	**Experiment No.:**	15	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Sum of Subset Problem						
2	Course Outcomes	Demonstrate Backtracking using Sumof Subset and Hamiltonian cycles.						
3	Aim	To find a **subset** of a given set **S** = {Sl, S2,.....,Sn} of ***n*** positive integers whose SUM is equal to a given positive integer ***d***.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Backtracking						

6	Procedure, Program, Activity, Algorithm, Pseudo Code	Algorithm:
		initialize a list *S* to contain one element 0.
		for each *i* from 1 to *N* do
		let *T* be a list consisting of *xi* + *y*, for all *y* in *S*
		let *U* be the union of *T* and *S*
		sort *U*
		make *S* empty
		let *y* be the smallest element of *U*
		add *y* to *S*
		for each element *z* of *U* in increasing order do
		//trim the list by eliminating numbers close to one another
		//and throw out elements greater than *s*
		if $y + cs/N < z \le s$, set $y = z$ and add *z* to *S*
		if *S* contains a number between $(1 – c)s$ and *s*, output *yes*, otherwise *no*						

7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output	Enter the size of the set: 5 Enter the set in increasing order: 1 2 5 6 8 Enter the required sum : 9 the solution to the sum of subset problem is: Subset1: 1 2 6 Subset2: 1 8						
9	Sample Calculations							
10	Graphs, Outputs							
11	Results & Analysis							
12	Application Areas							

13	Remarks	
14	Faculty Signature with Date	

Experiment 12: Hamiltonian Cycles using backtracking principle

-	**Experiment No.:**	15	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Hamiltonian Cycles						
2	Course Outcomes	Demonstrate Backtracking using Sumof Subset and Hamiltonian cycles.						
3	Aim	Design and implement in Java to find all Hamiltonian Cycles in a connected undirected Graph G of *n* vertices using backtracking principle.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Backtracking						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	***Algorithm:*** *Input:* A 2D array graph[V][V] where V is the number of vertices in graph and graph[V][V] is adjacency matrix representation of the graph. A value graph[i][j] is 1 if there is a direct edge from i to j, otherwise graph[i][j] is 0. *Output:* An array path[V] that should contain the Hamiltonian Path. path[i] should represent the ith vertex in the Hamiltonian Path. The code should also return false if there is no Hamiltonian Cycle in the graph.						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output	Enter No. of Vertices: 6 Enter No. of Edges: 9 Enter the Edge1: 1 2 Enter the Edge2: 1 3 Enter the Edge3: 1 4 Enter the Edge4: 2 3 Enter the Edge5: 2 6 Enter the Edge6: 3 4 Enter the Edge7: 3 5 Enter the Edge8: 5 6 Enter the Edge9: 4 5 Hamiltonian Cycle 1-->2-->6-->5-->3-->4-->1 1-->2-->6-->5-->4-->3-->1 1-->3-->2-->6-->5-->4-->1						
9	Sample							

	Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	
13	Remarks	
14	Faculty Signature with Date	

Experiment 01 : Structure of C program

-	**Experiment No.:**	1	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Structure of C program						
2	Course Outcomes	Design the structure of C program						
3	Aim	Exercise on structure of C program						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Basic structure of c program to writing the c program						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	• step 1: start • step 2: write programming • step 3: save the program • step 4: compile • step 5:if error then correct the errors • step 6:run • step 7:stop						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	• - • - • -						
8	Observation Table, Look-up Table, Output	• well come to jpnce • this is the first program in cp lab						
9	Sample Calculations	• - • - • -						
10	Graphs, Outputs	• - • -						
11	Results & Analysis	• - • -						
12	Application Areas	• To write the c program						
13	Remarks							
14	Faculty Signature with Date							

Experiment 02 : Keywords and identifiers

-	**Experiment No.:**	1	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Keywords and identifiers						
2	Course Outcomes	Design the logic for a given problem						
3	Aim	Exercise on Keywords and identifiers						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	To identify the key words in c programming To identify the identifiers in c programming						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	Step 1: start Step 2: read a,b Step 3: initialize the a,b Step 4: perform the operation in a,b Step 5: print the result step 6: stop						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							

8	Observation Table, Look-up Table, Output	Enter any 2 number 5,6 The sum of two variables 11
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	In searching and sorting concepts in data-structures and python
13	Remarks	
14	Faculty Signature with Date	

F. Content to Experiment Outcomes

1. TLPA Parameters

Table 1: TLPA – Example Course

Expt-#	Course Content or Syllabus (Split module content into 2 parts which have similar concepts)	Content Teaching Hours	Blooms' Learning Levels for Content	Final Blooms' Level	Identified Action Verbs for Learning	Instruction Methods for Learning	Assessment Methods to Measure Learning
A	*B*	*C*	*D*	*E*	*F*	*G*	*H*
1	Write a C++ program to read series of names, one per line, from standard input and write these names spelled in reverse order to the standard output using I/O redirection and pipes. Repeat the exercise using an input file specified by the user instead of the standard input and using an output file specified by the user instead of the standard output.	3	- L2 - L3 - L4	L4	- -	- Lecture - -	- Slip Test - -
2	Write a C++ program to read and write student objects with fixed length records and the fields delimited by "|". Implement pack ( ), unpack ( ), modify ( ) and search ( ) methods.	3	- L2 - L3 - L4	L4	- -	- Lecture - Tutorial -	- Assignment - -
3	Write a C++ program to read and write student objects with Variable - Length records using any suitable record structure. Implement pack ( ), unpack ( ), modify ( ) and search ( ) methods.	3	- L2 - L3 - L4	L4	- -	- Lecture -	- Assignment -
4	Write a C++ program to write student objects with Variable - Length records using any suitable record structure and to read from this file a student record using RRN.	3	- L2 - L3 - L4	L4	- -	- Lecture -	- Slip Test -
5	Write a C++ program to implement simple index on primary key for a file of student objects. Implement add ( ), search ( ), delete ( ) using the index.	3	- L2 - L3 - L4	L4	- -	- Lecture -	- Slip Test -
6	Write a C++ program to implement index on secondary key, the name, for a file of student objects. Implement add ( ), search ( ), delete ( ) using the secondary index.	3	- L2 - L3 - L4	L4	- -	- Lecture - Tutorial -	- Assignment - -
7	Write a C++ program to read two lists of names and then match the names in the	3	- L2 - L3	L4	- -	- Lecture	- Assignment

	two lists using Co Sequential Match based on a single loop. Output the names common to both the lists.		- L4			- Tutorial -	- -
8	Write a C++ program to read k Lists of names and merge them using k-way merge algorithm with k = 8.	3	- L2 - L3 - L4	L4	- -	- Lecture - Tutorial -	- Assignment - -
9	Student should develop mini Project on the topics mentioned below or similar applications Document processing, transaction management, indexing and hashing, buffer management, configuration management. Not limited to these.	3	- L2 - L3 - L4	L4	- -	- Lecture - -	- Assignment - -

2. Concepts and Outcomes:

Table 2: Concept to Outcome – Example Course

Expt - #	Learning or Outcome from study of the Content or Syllabus	Identified Concepts from Content	Final Concept	Concept Justification (What all Learning Happened from the study of Content / Syllabus. A short word for learning or outcome)	CO Components (1.Action Verb, 2.Knowledge, 3.Condition / Methodology, 4.Benchmark)	Course Outcome **Student Should be able to ...**
A	*I*	*J*	*K*	*L*	*M*	*N*
1	- -	- -	Klystron oscillator	Comprehend the working of Klystron oscillator	- Understand - Klystron Oscillator - -	Understand the working of Klystron Oscillator.
2	- -	- -	Microwave transmission lines	Examine the transmission lines using graphical methods	- Analyze - Transmission Lines - Graphical Methods -	Analyze the transmission lines using Graphical methods.
3	- -	- -	Multiport networks	Implement the Z, Y and S parameters to Multiport networks	- Analyze - Multiport Networks - -	Analyze the Z, Y and S parameters for a Multiport network.
4	- -	- -	Microwave passive devices	Understand the working of microwave passive devices	- Understand - Microwave Passive Devices - -	Understand the working of different microwave passive devices.
5	- -	- -	Striplines	Have knowledge of micro, parallel and shielded striplines	- Understand - Types of Stripline - -	Understand micro, parallel and shielded striplines.
6	- -	- -	Antenna parameters	Compute the antenna design characteristics using the parameters	- Apply - Design Characteristics - -	Describe antenna working using the given parameters.
7	- -	- -	Array of point sources	Extend the antenna parameters to the array of point sources	- Apply - Array of Point Sources - -	Describe the working of point sources.
8	- -	- -	Electric dipole antennas	Examine the field parameters of electric dipole antennas	- Analyze - Electric Dipole Antenna	Analyze the working of electric dipole antenna.
9	- -	- -	Loop and horn	Explain the working of horn and loop	- Understand - Horn and Loop	Explain the working of horn and loop

			antennas	antennas	Antenna	antennas.