Ref No:

SRI KRISHNA INSTITUTE OF TECHNOLOGY,BANGALORE

LABORATORY PLAN

Academic Year 2019-20

Program:	B E – Computer Science & Engineering
Semester :	6
Course Code:	17CSL67
Course Title:	SYSTEM SOFTWARE AND OPERATING SYSTEM LABORATORY
Credit / L-T-P:	2 / 0-1-2
Total Contact Hours:	40
Course Plan Author:	

Academic Evaluation and Monitoring Cell

No. 29, Chimny hills, Hesarghtta Road, Chikkabanavara
Bangalore – 560090, Karnataka, India
Phone/Fax: +91-08023721315/ 23721477
www.skit.org.in

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 :	3 /6	*Academic Year:*	2019-20
Course Title:	SYSTEM SOFTWARE AND OPERATING SYSTEM LABORATORY	*Course Code:*	17CSL67
Credit / L-T-P:	2 / 0-1-2	*SEE Duration:*	180 Minutes
Total Contact Hours:	40 Hrs	*SEE Marks:*	60 Marks
CIA Marks:	40 Marks	*Assignment*	-
Lab. Plan Author:	NAGARATHNA C	*Sign*	Dt :
Checked By:		*Sign*	Dt :

2. Laboratory Content

Expt.	Title of the Experiments	Lab Hours	Concept	Blooms Level

1	a)Write a LEX program to recognize valid arithmetic expression. Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately. b)Write YACC program to evaluate arithmetic expression involving operators: +, -, *, and /	3	Lexical analysis	L5
2	Develop, Implement and Execute a program using YACC tool to recognize all strings ending with ***b*** preceded by ***na's*** using the grammar $a^n b$ (note: input ***n*** value)	3	Parsing operation	L5
3	Design, develop and implement YACC/C program to construct ***Predictive /LL(1) Parsing Table*** for the grammar rules: ***A →aBa , B →bB | ε***. Use this table to parse the sentence: ***abba$***	3	Context free grammar	L5
4	Design, develop and implement YACC/C program to demonstrate ***Shift Reduce Parsing*** technique for the grammar rules: ***E →E+T | T, T →T*F | F, F →(E) | id*** and parse the sentence: ***id + id * id***.	3	Context free grammar	L5
5	Design , develop and implement a C/Java program to generate the machine code using triples for the statement ***A = -B * (C +D)*** whose intermediate code in three-address form: ***T1 = -B*** ***T2 = C + D*** ***T3 = T1 + T2*** ***A = T3***	3	Context free grammar	L5
6	a) Write a LEX program to eliminate ***comment lines*** in a ***C*** program and copy the resulting program into a separate file. b) Write YACC program to recognize valid ***identifier, operators and keywords in the given text (C program) file.***	3	Lexical analysis	L5
7	Design, develop and implement a C/C++/Java program to simulate the working of Shortest remaining time and Round Robin (RR) scheduling algorithms.Experiment with different quantum sizes for RR algorithm.	3	Process Scheduling	L5
8	Design, develop and implement a C/C++/Java program to implement Banker's algorithm. Assume suitable input required to demonstrate the results.	6	Deadlock handling	L5
9	Design, develop and implement a C/C++/Java program to implement page replacement algorithms LRU and FIFO. Assume suitable input required to demonstrate the results.	6	Memory management	L5
10	a) Design, develop and implement a C/C++/Java program to simulate a ***numerical calculator*** b) Design, develop and implement a C/C++/Java program to simulate ***page replacement technique***	5	Memory management	L5

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.)**	**-**	**-**
7,8,9,10	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Principles 7th edition, Wiley-India, 2006.		In Lib / In Dept
1,2,3,4,	System Software by Leland. L. Beck, D Manjula, 3rdedition, 2012		In Lib/ In dept

6			
5	Compilers-Principles, Techniques and Tools by Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman. Pearson, 2nd edition, 2007		
B	**Reference books (Title, Authors, Edition, Publisher, Year.)**	**-**	**-**
7,8,9,10	Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6thEdition	?	In Lib
7,8,9,10	D.M Dhamdhere, Operating Systems: A Concept Based A pproach 3rd Ed, McGrawHill, 2013	?	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			
?			

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	18CSP27	C Programming		1	-	L3 Apply
1,2,3,4,5,6	15CS44	Microprocessor and Microcontroller	Assembly Level Programming	4	-	Understand L2 Apply L3
-						
-						

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
1,2,3,4,5,6	Assembler Simulation	Placement, Entrepreneurship	A seminar on Assembler Simulation	Understand L2 Apply L3
1,2,3,4,5,6	Loader simulation	Placement, Entrepreneurship	A seminar on Loader simulation	Understand L2 Apply L3
1,2,3,	Lexical Analyser Implementation	Placement,	A seminar on Lexical Analyser	Understa

4,5,6		Entrepreneurship	Implementation	nd L2 Apply L3
1,2,3, 4,5,6	Parser Implementation	Placement, GATE, Higher Study, Entrepreneurship	A seminar on Parser Implementation	Understand L2 Apply L3
7,8,9	Deadlock detection algorithms	Higher Study	Gap A seminar on detection algorithms	Analyze L4
7,8,9	Design principles of Ubuntu OS	Higher Study	Gap A seminar on Ubuntu OS	Apply L3
-				

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, should disconnect the setup made by them, and should return all the components/instruments taken for the purpose.	
9	Any damage of the equipment or burn-out components 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, program code along with comments and output for various inputs given	

2. Laboratory Specific Instructions

SNo	Specific Instructions	Remarks
1	Opening the terminal	
2	Create a file with vi editor using command vi filename.l/ filename.y/filename.c	
3	write the lex/yacc/c program in a file and save and quit using the command Esc+:+Shiftkey+wq	
4	Compilation commands: • cc lex.yy.c -ll for lex program • cc lex.yy.c y.tab.h -ll for yacc programs • cc filename.c	
5	For running the program : ./a.out	
6	Compile the program F9	
7	Execute the program F10	

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	17CSL67.1	Demonstrate how the complier tools works & will be able to observe the use of Context free grammar to match pattern & tokenize the input	10	Lexical analysis	Demonstrate	Slip Test, viva	L5 Evaluate
2	17CSL67.2	Demonstrate how the complier tools YACC works & will be able to observe the parsing operations	10	Parsing operation	Demonstrate	Assignment, viva	L5 Evaluate
3	17CSL67.3	Demonstrate the operating system functions such as scheduling, multi threading & deadlock handling	10	Operating system concepts	Demonstrate	Assignment and Slip Test, viva	L5 Evaluate
4	17CSL67.4	To acquire the implementation knowledge of System Software concepts and FAFL grammar concepts through UNIX supported tools Lex and Yacc	10	Regular Expression	Demonstrate	Assignment, viva	L5 Evaluate
-		**Total**	**40**	**-**	**-**	**-**	**-**

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

2. Laboratory Applications

Expt.	Application Area	CO	Level
1	Lexers and parsers are most often used for compilers, but can be used for other computer language tools.	CO1	L5
2	**Parsers** are **used** everywhere in software. An XML **Parser** is a **parser** that is designed to read XML and create a way for programs to use XML.	CO2	L5
3	An **operating system** is the most important software that runs on a computer. It manages the computer's memory and processes, as well as all of its software and hardware.	CO3	L5
4	Context free grammar can be used to perform all types of text search and text replace operations. Context free grammar (REGEX) are language default functions (more of an expression) used for things like pattern **matching**, sorting, searching and many others.	CO4	L5

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'**	**-**
	CO1	PO1		'Knowledge of lexical analysis phases of Complier Design is required to implement programs using LEX tools	L2
		PO2		Analyzing different lexical programs requires knowledge of lexical analysis phases of Complier Design	L2
		PO3		Developing LEX programs requires the knowledge of complier desgin	L3
		PO4		Examine the programs developed	L4
		PO5		Modern tools usage to examine the developed programs	L5

		PO12		Learning in the context of technology changes	L5
	CO2	PO1		Knowledge of syntax analysis phases of Complier Design is required to implement programs using YACC tools	L2
		PO2		Analyzing programs written using YACC.	L2
		PO3		Design & develop new programs using YACC syntax and semantics.	L3
		PO4		Examine the programs developed	L4
		PO5		Modern tools usage to examine the developed programs	L5
		PO12		Learning in the context of technology changes	L5
	CO3	PO1		Knowledge of Operating system is required to implement programs for memory management and process management	L2
		PO2		Analyzing programs written for process management and deadlock handling	L2
		PO3		Design & develop new programs for scheduling, allocation and communication used in operating system	L3
		PO4		Examine the programs developed	L4
		PO12		Learning in the context of technology changes	L5
	CO4	PO1		Knowledge of context free grammar is required to implement programs for pattern matching.	L2
		PO2		Analyzing programs written for pattern matching	L2
		PO3		Design & develop new programs using LEX and YACC Y tools for solving different pattern matching algorithms	L3
		PO4		Examine the programs developed.	L4
		PO5		Modern tools usage to examine the developed programs	L5
		PO12		Learning in the context of technology changes	L5

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	Level
	17CSL67.1	Demonstrate how the complier tools works & will be able to observe the use of Context free grammar to match pattern & tokenize the input	3	3	3	2	2	-	-	-	-	-	-	3				L5
	17CSL67.2	Demonstrate how the complier tools YACC works & will be able to observe the parsing operations	3	3	3	2	3	-	-	-	-	-	-	3				L5
	17CSL67.3	Demonstrate the operating system functions such as scheduling, multi threading & deadlock handling	3	3	3	2	-	-	-	-	-	-	-	3				L5
	17CSL67.4	To acquire the implementation knowledge of System Software concepts and FAFL grammar concepts through UNIX supported tools Lex and Yacc	3	3	3	3	3	-	-	-	-	-	-	3				L5
-		**Average attainment (1, 2, or 3)**	**3**	**3**	**3**	**2**	**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	Deadlock detection algorithms	Extra class	Extra class		
2	Design principles of Ubuntu OS	Extra class	Extra class		
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					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					

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	a)Write a LEX program to recognize valid arithmetic expression. Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately. b)Write YACC program to evaluate arithmetic expression involving	03	1	-	1	-	-	-	1	CO	L5

	operators: +, -, *, and /										
2	Develop, Implement and Execute a program using YACC tool to recognize all strings ending with ***b*** preceded by ***na's*** using the grammar $a^n b$ (note: input ***n*** value)	03	1	-	1	-	-	-	1	CO	L5
3	Design, develop and implement YACC/C program to construct ***Predictive / LL(1) Parsing Table*** for the grammar rules: ***A →aBa , B →bB | ε***. Use this table to parse the sentence: ***abba$***	03	1	-	1	-	-	-	1	CO	L5
4	Design, develop and implement YACC/C program to demonstrate ***Shift Reduce Parsing*** technique for the grammar rules: ***E →E+T | T, T →T*F | F, F →(E) | id*** and parse the sentence: ***id + id * id***.	03		1	1	-	-	-	1	CO	L5
5	Design , develop and implement a C/Java program to generate the machine code using triples for the statement ***A = -B * (C +D)*** whose intermediate code in three-address form: ***T1 = -B*** ***T2 = C + D*** ***T3 = T1 + T2*** ***A = T3***	03		1	1	-	-	-	1	CO	L5
6.	a) Write a LEX program to eliminate ***comment lines*** in a ***C*** program and copy the resulting program into a separate file. b) Write YACC program to recognize valid ***identifier, operators and keywords in the given text (C program) file.***	03		1	1				1	CO	L5
7.	Design, develop and implement a C/C++/Java program to simulate the working of Shortest remaining time and Round Robin (RR) scheduling algorithms.Experiment with different quantum sizes for RR algorithm.	03			1				1	CO	L5
8.	Design, develop and implement a C/C++/Java program to implement Banker's algorithm. Assume suitable input required to demonstrate the results.	04			1				1	CO	L5
9.	Design, develop and implement a C/C++/Java program to implement page replacement algorithms LRU and FIFO. Assume suitable input required	04			1				1	CO	L5

	to demonstrate the results.										
10.	a) Design, develop and implement a C/C++/Java program to simulate a ***numerical calculator*** b) Design, develop and implement a C/C++/Java program to simulate ***page replacement technique***	06			1				1	CO	L5
-	**Total**	**40**	**3**	**3**	**10**	**-**	**-**	**-**	**80**	**-**	**-**

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	20	CO1, CO2,	L5
CIA Exam – 2	20	CO3, CO4 ,	L5
CIA Exam – 3	20	CO1, CO2,CO3, CO4 ,CO5	L5
Assignment - 1			
Assignment - 2			
Assignment - 3			
Seminar - 1			
Seminar - 2			
Seminar - 3			
Other Activities – define – Slip test			
Final CIA Marks	**20**	**-**	**-**

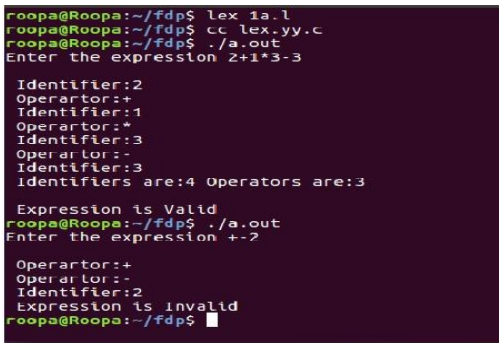
-

SNo	Description	Marks
1	Observation and Weekly Laboratory Activities	05 Marks
2	Record Writing	03 Marks for each Expt
3	Internal Exam Assessment	12 Marks
4	Internal Assessment	20 Marks
5	SEE	80 Marks
-	**Total**	**100 Marks**

E. EXPERIMENTS

Experiment 01 : Arithmetic Expression

-	**Experiment No.:**	1	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	a)Write a LEX program to recognize valid arithmetic expression. Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately. b)Write YACC program to evaluate arithmetic expression involving operators: +, -, *, and /						
2	Course Outcomes	Demonstrate how the complier tools works & will be able to observe the use of Context free grammar to match pattern & tokenize the input						
3	Aim	Program to recognize a valid arithmetic expression and to recognize the						

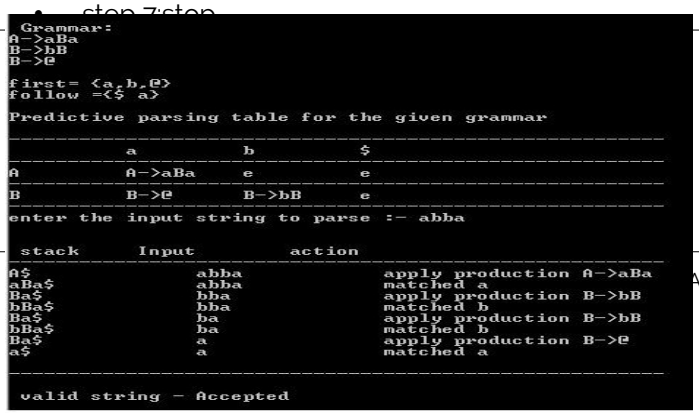
		identifiers and operators present. Print them separately.Exercise on structure of C program
4	Material / Equipment Required	Lab Manual
5	Theory, Formula, Principle, Concept	
6	Procedure, Program, Activity, Algorithm, Pseudo Code	• step 1: Start • step 2: write a program to validate the arithmetic expression • step 3: save the program • step 4: execute aprogram • 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	• the output of arithmetic expression is retrived
9	Sample Calculations	• - • - • -
10	Graphs, Outputs	• - • -
11	Results & Analysis	• - • -
12	Application Areas	Lexers and parsers are most often used for compilers, but can be used for other computer language tools.
13	Remarks	
14	Faculty Signature with Date	

Experiment 02 : String Recognization

-	**Experiment No.:**	2	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Develop, Implement and Execute a program using YACC tool to recognize all strings ending with ***b*** preceded by ***na's*** using the grammar $a^n b$ (note: input ***n*** value)						
2	Course Outcomes	Demonstrate how the complier tools YACC works & will be able to observe the parsing operations						
3	Aim	Program to recognize a valid variable, which starts with a letter b, followed by na's number of letters						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept							
6	Procedure,	• step 1: Start						

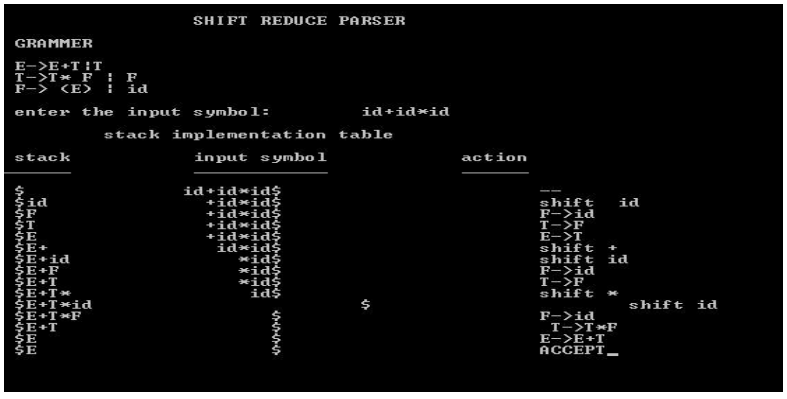
	Program, Activity, Algorithm, Pseudo Code	• step 2: write a program to recognizes trings • step 3: save the program • step 4: execute a program • step 5: if error then correct the errors • step 6:run • step 7:stop
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	
8	Observation Table, Look-up Table, Output	Recognizethe string with b and preceding with ***na's*** using the grammar ***a^n b***
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	**Parsers** are **used** everywhere in software. An XML **Parser** is a **parser** that is designed to read XML and create a way for programs to use XML.
13	Remarks	
14	Faculty Signature with Date	

Experiment 03 : Predictiv/LL(1) Parsing Table

-	**Experiment No.:**	3	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Design, develop and implement YACC/C program to construct ***Predictive /LL(1) Parsing Table*** for the grammar rules: ***A →aBa , B →bB |*** ε. Use this table to parse the sentence: ***abba$***						
2	Course Outcomes	Demonstrate how the complier tools YACC works & will be able to observe the parsing operations						
3	Aim	Program to implement predictive /LL(1) parsing table for grammar rule ***A →aBa , B →bB |*** ε						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code	• step 1: Start • step 2: write a program to for parsing table • step 3: save the program • step 4: execute a program • step 5: if error then correct the errors • step 6:run • step 7:stop						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							

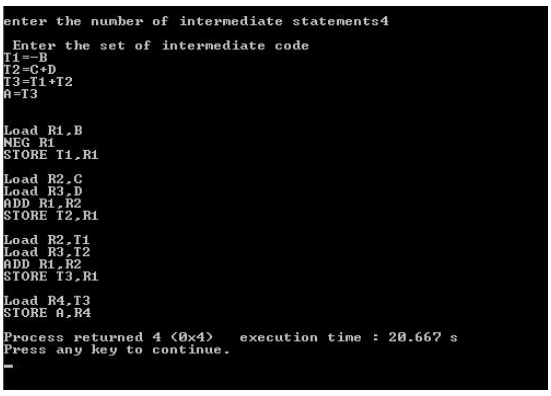
8	Observation Table, Look-up Table, Output	
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	**Parsers** are **used** everywhere in software. An XML **Parser** is a **parser** that is designed to read XML and create a way for programs to use XML.
13	Remarks	
14	Faculty Signature with Date	

Experiment 04 : Shift Reduce Parsing

-	**Experiment No.:**	3	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Design, develop and implement YACC/C program to demonstrate ***Shift Reduce Parsing*** technique for the grammar rules: ***E →E+T | T, T →T*F | F, F →(E) | id*** and parse the sentence: ***id + id * id***.						
2	Course Outcomes	Demonstrate how the complier tools YACC works & will be able to observe the parsing operations						
3	Aim	Program to demonstrate Shift Reduce Parsing						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code	• step 1: Start • step 2: write a program using shift reduce parsing technique for grammar rules: ***E →E+T | T, T →T*F | F, F →(E) | id*** and parse the sentence: ***id + id * id***. • step 3: save the program • step 4: execute a program • step 5: if error then correct the errors • step 6:run • step 7:stop						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							

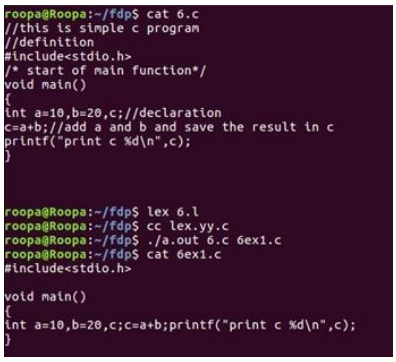
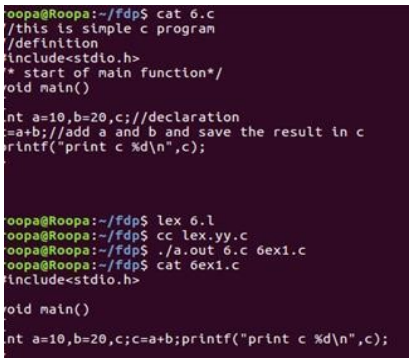
8	Observation Table, Look-up Table, Output	
9	Sample Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	**Parsers** are **used** everywhere in software. An XML **Parser** is a **parser** that is designed to read XML and create a way for programs to use XML.
13	Remarks	
14	Faculty Signature with Date	

Experiment 05: Generate the machine code using triples

-	**Experiment No.:**	3	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Design , develop and implement a C/Java program to generate the machine code using triples for the statement ***A = -B * (C +D)*** whose intermediate code in three-address form: ***T1 = -B ,2 = C + D, T3 = T1 + T2, A = T3***						
2	Course Outcomes	To acquire the implementation knowledge of System Software concepts and FAFL grammar concepts through UNIX supported tools Lex and Yacc						
3	Aim	Program to generate the machine code using triples for the statement ***A = -B * (C +D)***						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code	• step 1: Start • step 2: write a program using triples for the statement ***A = -B * (C +D)*** • step 3: save the program • step 4: execute a program • 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							
9	Sample Calculations							
10	Graphs, Outputs							
11	Results & Analysis							
12	Application Areas	Context free grammar can be used to perform all types of text search and text						

		replace operations. Context free grammar (REGEX) are language default functions (more of an expression) used for things like pattern **matching**, sorting, searching and many others.
13	Remarks	
14	Faculty Signature with Date	

Experiment 06 : Eliminate comment lines and recognize valid identifier ,operators and keyboards

-	**Experiment No.:**	3	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	a) Write a LEX program to eliminate ***comment lines*** in a ***C*** program and copy the resulting program into a separate file. b) Write YACC program to recognize valid ***identifier, operators and keywords in the given text (C program) file.***						
2	Course Outcomes	• Demonstrate how the complier tools works & will be able to observe the use of Context free grammar to match pattern & tokenize the input • Demonstrate how the complier tools YACC works & will be able to observe the parsing operations						
3	Aim	Program to eliminate comment lines and recognize valid identifier, operators and keywords						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code	• step 1: Start • step 2: write a program eliminate comment lines and recognize valid identifier, operators and keywords • step 3: save the program • step 4: execute a program • 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							
9	Sample Calculations							
10	Graphs, Outputs							
11	Results & Analysis							
12	Application Areas	• Lexers and parsers are most often used for compilers, but can be used for other computer language tools. • **Parsers** are **used** everywhere in software. An XML **Parser** is a **parser** that is designed to read XML and create a way for programs to use XML.						

13	Remarks	
14	Faculty Signature with Date	

Experiment 07 : Round Robin Scheduling Algorithm

-	**Experiment No.:**	3	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Design, develop and implement a C/C++/Java program to simulate the working of Shortest remaining time and Round Robin (RR) scheduling algorithms.Experiment with different quantum sizes for RR algorithm.						
2	Course Outcomes	Demonstrate the operating system functions such as scheduling, multi threading & deadlock handling						
3	Aim	Program to implement working of Shortest remaining time and Round Robin (RR) scheduling algorithms.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept	Create view table to solve the database queries						
6	Procedure, Program, Activity, Algorithm, Pseudo Code	Step1: Start the process Step2: Get the number of elements to be inserted Step3: Get the value for burst time for individual processes Step4: Get the value for time quantum Step5: Make the CPU scheduler go around the ready queue allocating CPU to each process for the time interval specified Step6: Make the CPU scheduler pick the first process and set time to interrupt after quantum. And after it's expiry dispatch the process Step7: If the process has burst time less than the time quantum then the process is released by the CPU Step8: If the process has burst time greater than time quantum then it is interrupted by the OS and the process is put to the tail of ready queue and the schedule selects next process from head of the queue Step9: Calculate the total and average waiting time and turn around time Step10: Display the results Step11: Stop the process						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	Enter the burst time and arrival time of process 11 4 1 Enter the burst time and arrival time of process 21 9 2 Enter the burst time and arrival time of process 31 5 3 Enter Time quantum: 4 p_id B_time A_Time 1 8 0 2 4 1 3 9 2 4 5 3 Quantum : 4 Processes Burst timeWaiting time Turn around time 1 812 20 2 43 7 3 915 24 4 517 22 Average waiting time = 11.75 Average turn around time = 18.25 8. Design, develop and implement a C/C++/Java program to implement Banker's Algorithm. Assume suitable input required to demonstrate the results						
8	Observation Table, Look-up Table, Output							
9	Sample Calculations							
10	Graphs, Outputs							
11	Results & Analysis							
12	Application Areas	An **operating system** is the most important software that runs on a computer. It manages the computer's memory and processes, as well as all of its software						

		and hardware.
13	Remarks	
14	Faculty Signature with Date	

Experiment 08 : Banker's Algorithm

-	**Experiment No.:**	3	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Design, develop and implement a C/C++/Java program to implement Banker's algorithm. Assume suitable input required to demonstrate the results.						
2	Course Outcomes	Demonstrate the operating system functions such as scheduling, multi threading & deadlock handling						
3	Aim	Design,develop and run a program to implement the Banker " s Algorithm. Demonstrate its Working with different data values.						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept							
	Procedure, Program, Activity, Algorithm, Pseudo Code	**Algorithm:** To check if a state is safe or not: i. Look for a row R whose count unmet resources needs are all smaller than A. if no such row exists; the system is deadlocked since no process can run to completion. • Assume the process of the row chosen requests all the resources it needs & finishes. Mark the process as terminated & add it's resources to A vector. ii. Repeat steps i & ii, until either all processes are marked terminated. If several processes are eligible to be chosen in step1, it does not matter which one is selected.							
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph							
8	Observation Table, Look-up Table, Output	Sample Out put: Enter the number of processes: 5 Enter the number of resources: 3 Enter available data 3 3 2 Enter max matrix 7 5 3 3 2 2 9 0 2 2 2 2 4 3 3 Enter allot matrix 0 1 0 2 0 0 3 0 2 2 1 1 0 0 2 Need --- 7 4 3 1 2 2 6 0 0 0 1 1 4 3 1 System is in safe state. Safe sequence is: 1 3 4 0 2							
9	Sample							

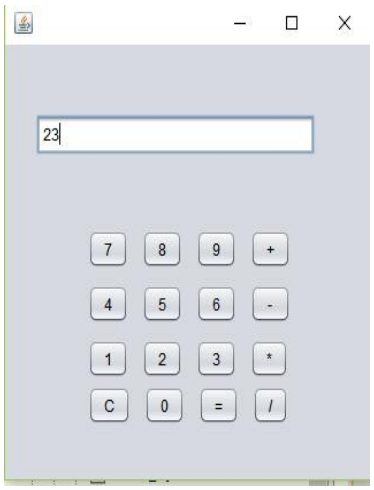
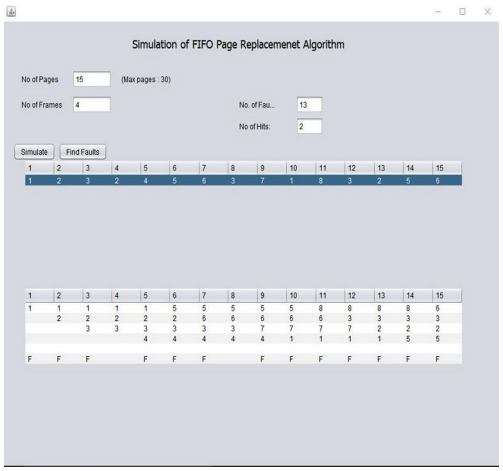
	Calculations	
10	Graphs, Outputs	
11	Results & Analysis	
12	Application Areas	An **operating system** is the most important software that runs on a computer. It manages the computer's memory and processes, as well as all of its software and hardware.
13	Remarks	
14	Faculty Signature with Date	

Experiment 09 : LRU and FIFO

-	**Experiment No.:**	3	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	Design, develop and implement a C/C++/Java program to implement page replacement algorithms LRU and FIFO. Assume suitable input required to demonstrate the results.						
2	Course Outcomes	Demonstrate the operating system functions such as scheduling, multi threading & deadlock handling						
3	Aim	Program toimplement placereplacement algorithms						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code	• step 1: Start • step 2: write a program to implement page replacement algorithms LRU and FIFO • step 3: save the program • step 4: execute a program • step 5: if error then correct the errors • step 6:run step 7:stop						
7	Block, Circuit, Model Diagram, Reaction Equation, Expected Graph	Page Replacement Algorithms 1.Enter data 2.FIFO 3.Optimal 4.LRU 5.LFU 6.Second Chance 7.Exit Enter your choice:1 Enter length of page reference sequence:8 Enter the page reference sequence:2 3 4 2 3 5 6 2 Enter no of frames:3 Page Replacement Algorithms 1.Enter data 2.FIFO 3.Optimal 4.LRU 5.LFU 6.Second Chance 7.Exit						
8	Observation Table, Look-up Table, Output							
9	Sample Calculations							
10	Graphs, Outputs							
11	Results & Analysis							

12	Application Areas	An **operating system** is the most important software that runs on a computer. It manages the computer's memory and processes, as well as all of its software and hardware.
13	Remarks	
14	Faculty Signature with Date	

Experiment 10 : Numerical Calculator and Page Replacement Technique.

-	**Experiment No.:**	3	**Marks**		**Date Planned**		**Date Conducted**	
1	Title	a) Design, develop and implement a C/C++/Java program to simulate a ***numerical calculator*** b) Design, develop and implement a C/C++/Java program to simulate ***page replacement technique***						
2	Course Outcomes	Demonstrate the operating system functions such as scheduling, multi threading & deadlock handling						
3	Aim	Program to implement numerical calculator and page replacement technique						
4	Material / Equipment Required	Lab Manual						
5	Theory, Formula, Principle, Concept							
6	Procedure, Program, Activity, Algorithm, Pseudo Code	• step 1: Start • step 2: write a program to implement numerical calculator andpage replacement. • step 3: save the program • step 4: execute a program • 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							
9	Sample Calculations							
10	Graphs, Outputs							
11	Results & Analysis							
12	Application Areas	An **operating system** is the most important software that runs on a computer. It manages the computer's memory and processes, as well as all of its software and hardware.						
13	Remarks							

14	Faculty Signature with Date	

F. Content to Experiment Outcomes

1. TLPA Parameters

Table 1: TLPA – 17CSL67 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	a)Write a LEX program to recognize valid arithmetic expression. Identifiers in the expression could be only integers and operators could be + and *. Count the identifiers & operators present and print them separately. b)Write YACC program to evaluate arithmetic expression involving operators: +, -, *, and /	3	- L2 - L3 - L5	L5	- Demonstrate -Evaluate	- Lecture - presentation -	- Slip Test - -
2	Develop, Implement and Execute a program using YACC tool to recognize all strings ending with ***b*** preceded by ***na's*** using the grammar $a^n b$ (note: input ***n*** value)	3	- L2 - L3 - L5	L5	Demonstrate -Evaluate	- Lecture - presentation -	- Assignment - -

3	Design, develop and implement YACC/C program to construct ***Predictive /LL(1) Parsing Table*** for the grammar rules: ***A →aBa , B →bB |*** ε. Use this table to parse the sentence: ***abba$***	3	- L2 - L3 - L5	L5	- Demonstrate -Evaluate	- Lecture - presentation	- Assignment -
4	Design, develop and implement YACC/C program to demonstrate ***Shift Reduce Parsing*** technique for the grammar rules: ***E →E+T | T, T →T*F | F, F →(E) | id*** and parse the sentence: ***id + id * id***.	3	- L2 - L3 - L5	L5	Demonstrate -Evaluate	- Lecture -	- Slip Test -
5	Design , develop and implement a C/Java program to generate the machine code using triples for the statement ***A = -B * (C +D)*** whose intermediate code in three-address form: ***T1 = -B*** ***T2 = C + D*** ***T3 = T1 + T2*** ***A = T3***	3	- L2 - L3 - L5	L5	Demonstrate -Evaluate	- Lecture - presentation	- Slip Test -
6	a) Write a LEX program to eliminate ***comment lines*** in a ***C*** program and copy the resulting program into a separate file. b) Write YACC program to recognize valid ***identifier, operators and keywords in the given text (C program) file.***	4	- L2 - L3 - L5	L5	- Demonstrate -Evaluate	- Lecture - presentation -	- Assignment - -
7	Design, develop and implement a C/C++/Java program to simulate the working of Shortest remaining time and Round Robin (RR) scheduling algorithms.Experiment with different quantum sizes for RR algorithm.	4	- L2 - L3 - L5	L5	- Demonstrate -Evaluate	- Lecture - presentation -	- Assignment - -
8	Design, develop and implement a C/C++/Java program to implement Banker's algorithm. Assume suitable input required to demonstrate the results.	4	- L2 - L3 - L5	L5	- Demonstrate -Evaluate	- Lecture - presentation -	- Assignment - -
9	Design, develop and implement a C/C++/Java program to implement page replacement algorithms LRU and FIFO. Assume suitable input required to demonstrate the results.	4	- L2 - L3 - L5	L5	- Demonstrate -Evaluate	- Lecture - presentation -	- Assignment - -
10	a) Design, develop and implement a C/C++/Java program to simulate a ***numerical calculator*** b) Design, develop and implement a C/C++/Java program to simulate ***page replacement technique***	6	- L2 - L3 - L5	L5	Demonstrate -Evaluate	presentation	

2. Concepts and Outcomes:

Table 2: Concept to Outcome – 17CSL67 Course

Expt - #	Learning or Outcome	Identified Concepts	Final Concept	Concept Justification	CO Components (1.Action Verb,	Course Outcome

	from study of the Content or Syllabus	from Content		(What all Learning Happened from the study of Content / Syllabus. A short word for learning or outcome)	2.Knowledge, 3.Condition / Methodology, 4.Benchmark)	**Student Should be able to ...**
A	*I*	*J*	*K*	*L*	*M*	*N*
1	-study of complier phase -lex tools	-complier phases - lexical analyser	Lexical analysis	Comprehend the working of Klystron oscillator	- implement - context free grammar - -	Demonstrate how the complier tools works & will be able to observe the use of Context free grammar to match pattern & tokenize the input
2	-study of parsing -token generations -YACC tools	-Parsing operation -	Parsing operation	Examine the transmission lines using graphical methods	- Evaluate - YACC tools	Demonstrate how the complier tools YACC works & will be able to observe the parsing operations
3	-study of deadlock states -operating system concepts	-deadlock handling -	Operating system concepts	Implement the Z, Y and S parameters to Multiport networks	- Evaluate - deadlock handling	Demonstrate the operating system functions such as scheduling, multi threading & deadlock handling
4	-Study of UNIX tools -CFC grammars	-context free grammar -regular expression	Regular Expression	Understand the working of microwave passive devices	- construct - FAFL grammar	To acquire the implementation knowledge of System Software concepts and FAFL grammar concepts through UNIX supported tools Lex and Yacc