

SYSTEM SOFTWARE LABORATORY (Effective from the academic year 2018 -2019) SEMESTER – VI			
Course Code	18CSL66	CIE Marks	40
Number of Contact Hours/Week	0:2:2	SEE Marks	60
Total Number of Lab Contact Hours	36	Exam Hours	03
Credits – 2			
Course Learning Objectives: This course (18CSL66) will enable students to: <ul style="list-style-type: none"> To make students familiar with Lexical Analysis and Syntax Analysis phases of Compiler Design and implement programs on these phases using LEX & YACC tools and/or C/C++/Java To enable students to learn different types of CPU scheduling algorithms used in operating system. To make students able to implement memory management - page replacement and deadlock handling algorithms 			
Descriptions (if any): Exercises to be prepared with minimum three files (Where ever necessary): <ol style="list-style-type: none"> Header file. Implementation file. Application file where main function will be present. The idea behind using three files is to differentiate between the developer and user sides. In the developer side, all the three files could be made visible. For the user side only header file and application files could be made visible, which means that the object code of the implementation file could be given to the user along with the interface given in the header file, hiding the source file, if required. Avoid I/O operations (printf/scanf) and use <i>data input file</i> where ever it is possible.			
Programs List:			
Installation procedure of the required software must be demonstrated, carried out in groups and documented in the journal.			
1.			
a.	Write a LEX program to recognize valid <i>arithmetic expression</i> . 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 <i>arithmetic expression</i> involving operators: +, -, *, and /		
2.	Develop, Implement and Execute a program using YACC tool to recognize all strings ending with <i>b</i> preceded by <i>n a</i> 's using the grammar <i>aⁿ b</i> (note: input <i>n</i> value)		
3.	Design, develop and implement YACC/C program to construct <i>Predictive / LL(1) Parsing Table</i> for the grammar rules: <i>A → aBa</i> , <i>B → bB ε</i> . Use this table to parse the sentence: <i>abba\$</i>		
4.	Design, develop and implement YACC/C program to demonstrate <i>Shift Reduce Parsing</i> technique for the grammar rules: <i>E → E+T T</i> , <i>T → T*F F</i> , <i>F → (E) id</i> and parse the sentence: <i>id + id * id</i> .		
5.	Design, develop and implement a C/Java program to generate the machine code using <i>Triples</i> for the statement <i>A = -B * (C +D)</i> whose intermediate code in three-address form: $T1 = -B$ $T2 = C + D$ $T3 = T1 + T2$ $A = T3$		

6.	
a.	Write a LEX program to eliminate <i>comment lines</i> in a <i>C</i> program and copy the resulting program into a separate file.
b.	Write YACC program to recognize valid <i>identifier, operators and keywords</i> in the given text (<i>C program</i>) file.
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.
8.	Design, develop and implement a C/C++/Java program to implement Banker's algorithm. Assume suitable input required to demonstrate the results
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.
Laboratory Outcomes: The student should be able to:	
<ul style="list-style-type: none"> • Implement and demonstrate Lexer's and Parser's • Evaluate different algorithms required for management, scheduling, allocation and communication used in operating system. 	
Conduct of Practical Examination:	
<ul style="list-style-type: none"> • Experiment distribution <ul style="list-style-type: none"> ○ For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity. ○ For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity. • Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only. • Marks Distribution (<i>Courseed to change in accordance with university regulations</i>) <ul style="list-style-type: none"> m) For laboratories having only one part – Procedure + Execution + Viva-Voce: 15+70+15 = 100 Marks n) For laboratories having PART A and PART B <ul style="list-style-type: none"> i. Part A – Procedure + Execution + Viva = 6 + 28 + 6 = 40 Marks ii. Part B – Procedure + Execution + Viva = 9 + 42 + 9 = 60 Marks 	