

20<sup>th</sup> July 2021

## B.Sc. (H) Computer Science V Semester (LOCF)

## System Programming (BHCS15B) Discipline Specific Elective - (DSE)

S.No.	Topic	Reference	Contents	Lectures
1	<b>Assemblers &amp; Loaders, Linkers:</b> One pass and two pass assembler, design of an assembler, Absolute loader, relocation and Linking concepts, relocating loader and Dynamic Linking	[2]	<b>Ch 3</b> (complete) [p36-62], <b>Ch 4</b> (complete) [p63- 83]	12L
2	<b>Introduction:</b> Overview of compilation, Phases of a compiler	[1]	<b>Ch 1</b> Sec 1.1-1.2[p1-12]	2L
3	<b>Lexical Analysis:</b> Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, Lexical Analyzer Generator	[1]	<b>Ch 3</b> Sec 3.1 [p109- 114], Sec 3.3-3.4 [p116-134] Sec 3.5[p140-145]	10L
4	<b>Parsing:</b> Bottom-up parsing-LR parser, Parser Generator- YACC	[1]	<b>Ch 4</b> Sec 4.1[Till 4.1.3][p192-195] Sec 4.2[Till 4.2.5][p197-203] Sec 4.4.2 FIRST and FOLLOW [p220- 222], Sec 4.5-4.7.4 [p233-270], Sec 4.9[Till 4.9.3] [p287-295]	14L
5	<b>Intermediate representations:</b> Three address code generation, syntax directed translation, translation of types, control statements	[1]	<b>Ch 5</b> Sec 5.1-5.2.4 [p303-314], <b>Ch 6</b> Sec 6.1[Till 6.1.1][p357-360] Sec 6.2[Till 6.2.3] [p363-369], Sec 6.3-6.3.4 [p370-375], Sec 6.4-6.4.1 [p378-380], Sec 6.4.3- 6.4.4 [p381- 384], Sec 6.5-6.5.2 [p386-390], Sec 6.6[Till 6.6.4] [p399-405], Sec 6.6.6 [p408] Sec 6.8-6.8.2 [p418-421]	12L
6	<b>Storage organization:</b> Activation records, stack allocation	[1]	<b>Ch 7</b> Sec 7.1-7.2 [p427-441]	5L
7	<b>Code Generation:</b> Object code generation	[1]	<b>Ch 8</b> Sec 8.1-8.3(upto 8.3.1) [p505-520]	5L

## Recommended Reading Material

### References

1. Aho, A., Lam, M., Sethi, R., & Ullman, J. D. (2006). *Compilers: Principles, Techniques, and Tools*. 2nd edition. Addison Wesley.
2. Chattopadhyaya, S. (2011). *System Software*. PHI Learning.

### Additional Resources:

1. Beck, L. & Manjula, D. (1996). *System Software: An Introduction to System Programming*. 3rd edition. Pearson Education.
2. Dhamdhere, D. M. (2015). *Systems Programming*. Tata McGrawHill.

### Programs to get familiar with Lex and Yacc

1. Write a Lex program to count the number of lines and characters in the input file.
2. Write a Lex program that implements the Caesar cipher: it replaces every letter with the one three letters after in alphabetical order, wrapping around at Z. e.g. a is replaced by d, b by e, and so on z by c.
3. Write a Lex program that finds the longest word (defined as a contiguous string of upper- and lower-case letters) in the input.
4. Write a Lex program that distinguishes keywords, integers, floats, identifiers, operators, and comments in any simple programming language.
5. Write a Lex program to count the number of identifiers in a C file.
6. Write a Lex program to count the number of words, characters, blank spaces and lines in a C file.
7. Write a Lex specification program that generates a C program which takes a string "abcd" and prints the following output.  
abcd  
abc  
ab  
a
8. A program in Lex to recognize a valid arithmetic expression.
9. Write a YACC program to find the validity of a given expression (for operators + - \* and /)
10. A Program in YACC which recognizes a valid variable which starts with letter followed by a digit. The letter should be in lowercase only.
11. A Program in YACC to evaluate an expression (simple calculator program for addition and subtraction, multiplication, division).
12. Program in YACC to recognize the strings "ab", "aabb", "aaabbb",... of the language ( $a^n b^n$ ,  $n \geq 1$ ).
13. Program in YACC to recognize the language ( $a^n b$ ,  $n \geq 10$ ). (Output to say input is valid or not)