UNIVERSITY OF PETROLEUM & ENERGY STUDIES

विद्याया शक्ति वशम्

SEMESTER VI

CSEG3015	Compiler Design	L	T	P	C
Version 1.0		**3**	**0**	**0**	**3**
Pre-requisites/Exposure	**Data Structures, knowledge of automata theory, basic knowledge of computer architecture**				
Co-requisites	**--**				

Course Objectives

1. To introduce the major concept areas of language translation and compiler design.
2. To enrich the knowledge in various phases of compiler and its use, code optimization techniques, machine code generation, and use of symbol table.
3. To extend the knowledge of parser by parsing LL parser and LR parser.
4. To provide practical programming skills necessary for constructing a compiler.

Course Outcomes
On completion of this course, the students will be able to

CO 1. Comprehend different phases of compiler.
CO 2. Useconcepts of regular grammar to build lexical analyzer.
CO 3. Build parsers for a context free grammar.
CO 4.Synthesize syntax directed translations rules.
CO 5.Assess code and memory optimization techniques to improve the performance of a program.

Catalog Description
This course explores the principles, algorithms, and data structures involved in the design and construction of compilers. Topics include finite-state machines, lexical analysis, context-free grammars, LR and LALR parsers, other parsing techniques, symbol tables, error recovery, and an introduction to intermediate code generation.
Course Content

UNIT I:Introduction **8 lecture hours**
Compiler, Phases and Passes, Bootstrapping, Finite State Machines and Regular Expressions and their Applications to Lexical Analysis, Implementation of Lexical Analyzers, Lexical Analyzer Generator, LEX, Formal Grammars and their Applications to Syntax Analysis, BNF Notation, Ambiguity, YACC. The Syntactic Specification of Programming Languages: Context Free Grammars, Derivation and Parse Tree, Capabilities of CFG.

UNIT II: Basic Parsing Techniques **10 lecture hours**
Parsers, Shift Reduce Parsing, Operator Precedence Parsing, Top Down Parsing, Predictive Parsing, Automatic Construction of Efficient Parsers: LR Parsers, The Canonical Collection of LR(0) items, Constructing SLR Parsing Tables, Constructing Canonical LR Parsing Tables, Constructing LALR Parsing Tables, Using Ambiguous Grammars, An Automatic Parser Generator, Implementation of LR Parsing Tables, Constructing LALR set of items

UNIT III: Syntax-Directed Translation **8 lecture hours**
Syntax Directed Translation Schemes, Implementation of Syntax Directed Translators, Intermediate Code, Postfix Notation, Parse Tree & Syntax Tree, Three Address Code, Quadruples & Triples, Translation of Assignment Statements, Boolean Expressions, Statements that alters the Flow of Control, Postfix Translation, Translation with a Top Down Parser, More about Translation: Array Reference in Arithmetic Expressions, Procedure Calls, Declaration, and Case Statements.

UNIT IV: Symbol Table **6 lecture hours**
Data Structures for Symbol Tables, Representing Score Information, Run Time Administration: Implementation of Simple Stack Allocation Scheme, Storage Allocation in Block Structures Language, Error Detection and Recovery: Lexical Phase Error, Syntactic Phase Errors, Semantic Phase Errors.

UNIT V:Introduction to Code Optimization **5 lecture hours**
Loop Optimization, the DAG Representation of Basic Blocks, Value Number and Algebraic Laws, Global Data-Flow Analysis

Text Books

1. Alfred V. Aho, Ravi Sethi Jeffrey D. Ullman, “Compilers- Principles, Techniques, and Tools”, 2nd Edition, Pearson Education Asia
2. Robin Hunter, “The Essence of Compiler”, 2nd Edition, Pearson Publication

Reference Books

1. Randy Allen, Ken Kennedy, “Optimizing Compilers for Modern Architectures: A Dependence-based Approach”, Morgan Kaufmann Publishers, 2002.
2. Steven S. Muchnick, “Advanced Compiler Design and Implementation, “Morgan Kaufmann Publishers - Elsevier Science, India, Indian Reprint 2003.
3. Keith D Cooper and Linda Torczon, “Engineering a Compiler”, Morgan Kaufmann Publishers Elsevier Science, 2004.
4. Charles N. Fischer, Richard. J. LeBlanc, “Crafting a Compiler with C”, Pearson Education, 2008.

Modes of Evaluation: Quiz/Assignment/ Presentation/ Extempore/ Written Examination
Examination Scheme:

Components	MSE	Presentation/Assignment/ etc	ESE
Weightage (%)	20%	30%	50%

Relationship between the Course Outcomes (COs), Program Outcomes (POs) and Program Specific Objectives (PSOs)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	3	-	-	-	-	-	-	-	-	-	2	-	-
CO2	2	2	3	-	1	-	-	-	-	-	-	-	2	-	-

CSEG3115	Compiler Design Lab	L	T	P	C
Version 1.0		0	0	2	1
Pre-requisites/Exposure	--				
Co-requisites	--				

Course Objectives

1. To comprehend different phases of a compiler.
2. Applying the concepts of regular grammar to build a lexical analyzer.
3. To build parsers for context-free grammar.
4. To design and develop syntax-directed translations rules.

Course Outcomes

Upon completion of this course, the learners will be able to:

CO1. Acquire knowledge in different phases and passes of Compiler, and specifying different types of tokens by lexical analyzer, and also able to use the Compiler tools like LEX, YACC, etc.

CO2. Develop the understanding of Parser and its types i.e. Top-down and Bottom-up parsers.

CO3. Construct LL, SLR, CLR, and LALR parse table.

CO4. Learn the techniques for code optimization.

Catalog Description

This course provides practical knowledge and hands-on experience of the design and implementation of different phases of the compiler. The lab experimental activities include analysis of code from lexical levels to syntactic levels, further the semantical analysis. The C-programming language is being used for the implementation of the lab experiments.

List of Experiments

EXPERIMENT NO. – 1

Title: Study of Lex and Yacc Tools

EXPERIMENT NO. – 2

Title: Introductory Problems using Lex Tool
List of Lab Activities:

1. WAP to count number of vowels and consonants in a given string.
2. WAP to count the number of characters, words, spaces, and end of lines in a given input file.
3. WAP to count number of comment lines in a given C program.

EXPERIMENT NO. – 3

Title: Lexical Analysis through Lex Tool
List of Lab Activities:

1. WAP to count number of '*scanf*' and '*printf*' statements in a C program. Replace them with '*read'* and '*write'* statements respectively.
2. WAP to recognize and count the number of identifiers in a given input file.
3. WAP to recognize a valid arithmetic expression and identify the identifiers and operators present. Print them separately.

EXPERIMENT NO. – 4

Title: Syntax Analysis via Yacc Tool
List of Lab Activities:

1. WAP to test the validity of a simple expression involving operators- +, -, *, and /.
2. WAP to evaluate an arithmetic expression involving operators- +, -, *, and /.

EXPERIMENT NO. – 5

Title: Simulation of FIRST & FOLLOW of a Grammar
.**Lab Activity:**
WAP toh simulate FIRST and FOLLOW of a grammar in C

EXPERIMENT NO. – 6

Title: Construction of LL(1) Parser
Lab Activity:
WAP to construct LL(1) parser for an expression in C

EXPERIMENT NO. – 7

Title: Design of a Predictive Parser for a given Language
Lab Activity:
WAP to implement a Predictive Parser for a given language in C

EXPERIMENT NO. – 8

Title: Calculation of Leading for all the Non-Terminals
Lab Activity:
WAP to calculate Leading for all the Non-Terminals in C

EXPERIMENT NO. – 9

Title: Calculation of Trailing for all the Non-Terminals
Lab Activity:
WAP to calculate Trailing for all the Non Terminals in C

EXPERIMENT NO. – 10

Title: Development of an Operator Precedence Parser
Lab Activity:
WAP to implement an Operator Precedence Parser for a given language in C

EXPERIMENT NO. – 11

Title: Implementation of Shift-Reduce Parser
Lab Activity:
WAP to implement Shift-Reduce Parser in C

EXPERIMENT No. – 12

Title: Design of a LALR Bottom-Up Parser for a given Language
Lab Activity:
WAP to derive a LALR Bottom Up Parser for the given language in C

Text Books

1. Alfred V. Aho, Monica S. Lam, Ravi Sethi, & Jeffrey D. Ullman, :Compilers- Principles, Techniques, and Tools".

Reference Books

1. Robin Hunter, "The Essence of Compilers".
2. John R. Levine, " lex & yacc".

Continuous Evaluation

There will be continuous evaluation for all practical subjects of SoCS during the semester. The performance of a student in a Practical subject will be evaluated as per component of evaluation given below:

- Viva voce / Quiz (50%)
- Performance & Records (50%).

Lab performance and record evaluation shall be a continuous process throughout the semester. Minimum two Viva-voce and two Quizzes based on practical sessions shall be conducted during the semester.

Relationship between Program Outcomes (POs), Program Specific Outcomes (PSOs) and Course Outcomes (COs)

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	-	1	2	1	-	-	-	-	-	2	3	-	-
CO2	1	1	-	1	2	1	-	-	-	-	-	2	3	-	-
CO3	1	1	-	1	2	1	-	-	-	-	-	2	3	-	-
CO4	1	1	-	1	2	1	-	-	-	-	-	2	3	-	-
Average	1	1	-	1	2	1	-	-	-	-	-	2	3	-	-

1=weak 2= moderate 3=strong