

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSC14	CC	Principles of Compiler Construction	3	0	2	4	15	15	40	15	15	Theory Automata and Formal Languages

#### COURSE OUTCOMES

- 1 Understand the internal organization and behavior of the compilers and other language processors.
- 2 Apply the formal constructs for designing a compiler.
- 3 Study and understand the functioning of a compiler.
- 4 Gain an ability to design simple domain-specific languages (DSLs) using compiler construction tools.
- 5 Go for the translation of languages or design the tools for online processing.

#### COURSE CONTENTS

##### Unit 1

**Introduction:** Language processors, structure of a compiler, compiler-construction tools, evolution of programming languages, applications of compiler technology, Transition diagrams, bootstrapping, just-in-time compilation.

##### Unit 2

**Lexical analysis:** Input buffering, specification and recognition of tokens, lexical analyzer generator.

##### Unit 3

**Syntax analysis:** Specification of syntax using grammar. Top-down parsing – recursive-descent, predictive. Bottom-up parsing – shift-reduce, SLR, CLR, LALR. Parser generator.

##### Unit 4

**Intermediate-code generation:** Syntax-directed translation. Three-address code. Translation of declarations, expressions, control flow. Backpatching. Runtime environment: Activation trees and records.

##### Unit 5

Code optimization: Sources of optimization, basic blocks, optimization of basic blocks, data-flow analysis, loop optimizations. Code generation: Issues, register allocation and assignment, peephole optimization

#### Practical:

1. Develop simple language processors like desk calculator and assembler.
2. Design a small high-level language.
3. Develop a lexical analyzer and a syntax analyzer for the same using the LEX and YACC tools. Also implement the bookkeeper module.
4. Design a small high-level language and implement a compiler for the same. If the target machine of the compiler is a hypothetical machine, then implement a simulator for it.
5. Develop a simple calculator using LEX and YACC tools.
6. Implement a program for symbol table using hashing
7. Implement a two-pass assembler

8. Implement a bottom-up parser using YACC tool.
9. Represent 'C' language using Context Free Grammar
10. Add assignment statement, If then else statement and while loop to the calculator and generate the three address code for the same.

#### SUGGESTED READINGS

1. Aho, A. V., Lam, M. S., Sethi, R. and Ullman J. D., "Compilers – Principles, Techniques and Tools (2nd ed.)", Pearson.
2. Chattopadhyay, S. 2005, "Compiler Design, PHI".
3. Appel, A. W. 200, "Modern Compiler Implementation in C", Cambridge University Press.
4. Kenneth C. Louden (1997), Compiler Construction– Principles and Practice, 1st edition, PWS Publishing.

Course No.	Type	Subject	L	T	P	Credits	CA	MS	ES	CA	ES	Pre-requisites
COCSC15	CC	Cloud Computing	3	1	0	4	25	25	50			Networking

#### COURSE OUTCOMES

1. Understand the concept of cloud computing, its quality issues, services, applications, benefits and limitations.
2. Understand the underlying technologies that drive a cloud computing environment.
3. To keep abreast of the trends in cloud technology
4. To get acquainted with available cloud environments such as GoogleApps, Microsoft Azure and Amazon Web Services.
5. To be able to use the cloud services.

#### COURSE CONTENT

##### UNIT I

**Introduction:** Concept of a cloud, Purpose, characteristics, challenges and developments in cloud computing, Virtualization, On-demand Cloud Computing, Current cloud Technologies and Environments, Benefits and limitations.

##### UNIT II

**Virtualization:** Characteristics of virtualization, Types of virtualization, Hypervisors and some case studies.

##### UNIT III

**Cloud architectures:** Software as a Service, Platform as a Service, Infrastructure as a Service, Storage as a Service, Applications as a Service, other services

##### UNIT IV

**Types of cloud architectures:** Public, Private, Hybrid, Design issues with cloud: scalability, fault tolerance, security, trust, privacy.